Issue Number: 1 **Alt. Numbers:** FRANCE-227-DIS-001 **Issue Date:** 23-Jan-98
Issue Title: Reference to Plib Dictionaries and Libraries in Clause 1
Issue Owner: G. Pierra (ENSMA), W. Du (EDF **Country:** France
Doc number SC4 N580 **Clause(s):** 1
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** Scope

Description:

External reference capabilities to Plib dictionaries and libraries is in scope.

AP 227 should support external reference to classes, properties, instances of standard parts and representation of standard parts that are stored in accordance with ISO 13584. In the current version of the document these external reference capabilities are not completely developed, they only appear through normative references to ISO 13584-24 and ISO 13584-42, and through the known_source mechanism in section 5.2.3.1.31. We suggest to state that these capabilities belong to the scope of AP 227.

Proposal **Date:** 23-Jan-98 **Proposer:** G. Pierra (ENSMA),

The following sentences should be added to the in-scope list:

- external reference to classes that are stored in accordance with ISO 13584
- external reference to properties that are stored in accordance with ISO 13584
- external reference to standard parts that are stored in accordance with ISO 13584
- external reference to representations ofstandard parts that are stored in accordance with ISO 13584

Comment **Date:** 03-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

Add 1st, 3rd, and 4th bullet of proposed solution to Scope without "that are stored in accordance with ISO 13584".

Issue Number: 2 **Alt. Numbers:** FRANCE-227-DIS-002 **Issue Date:** 23-Jan-98
Issue Title: Reference to PLib Dictionaries and Libraries
Issue Owner: G. Pierra (ENSMA), W. Du (EDF **Country:** France
Doc number SC4 N580 **Clause(s):** 4.2, 4.2.13
Issue status: rejected **Classification** Minor Technical
Source: DIS Balloting **Subject:** Integration

Description:

Support of external reference capabilities to Plib dictionaries and libraries.

AP 227 should support external reference to classes, properties, instances of standard parts and representation of standard parts that are stored in accordance with ISO 13584. In the current version of the document these external reference capabilities are not completely developed. We prcise below the different kinds of external reference capabilities that should be supported and discuss how they should be defined at the ARM level. Regarding the AIM level, we sugest to follow the solution developed for APs 214 and 212 in order to promote AP inter-operability.

1. Capability to reference a Plib-defined class (for classifying in an externally_defined_classification)

Information requirements:
supplier_BSU = 18 characters code
+ class_code = 14 characters code
+ class_version = 3 characters code
We call this set of information a class_BSU.

2. Capability to reference a Plib-defined property (for e.g., associated it with a product)

Information requirements:
class_BSU (above)
+ property_code = 14 characters code
+ property_version = 3 characters code
We call this set of information a property_BSU.

3. Capability to identify an externally_defined_plant_item within product data as a Plib-library-defined part.

Information requirements:
1 - one class_BSU (above
2 - a set of couple (property_BSU, value)
3 - an association that characterizes the fact that the above information IDENTIFIES the part.

Note that the information requirments defined in clause 4.2.13 for a catalogue_item are not sufficient to identify an item in a Plib compliant library.

As currently defined in AP 227, a catalogue is an explicit set of items. This set is defined by extension, therefore each item exists and may be associated with an item_name. Plib is mainly intended to define implicitly (intentionally) a set of items by gathering them in a class and by identifying each of the (implicit) members by means of the value of one or several properties.

Both kinds of << catalogues >> do exist in real life and it should be possible to reference both kinds from AP 227.

Two solutions might be considered:
1) to consider the existing catalogue definition in AP 227 as a special case of the more general definition from PLib (a catalogue_item is identified by THREE particular properties that are the item_name, item_version, and model_number)
2) to introduce two application objects, that might be called catalogue_item and ISO_13584_library_item (or library_item) for the two different kinds of << catalogues >>.

The second solution would probably be simpler and better from an AP inter-operability point of view.

4. Capability to express that a representation (of, e.g., an externally_defined_plant_item) is externally-defined, and that it should be regenerated from some PLib library

Information requirements:
(1) - The externally-defined representation shall contain an axis2-placement for positioning and orienting the representation (to be used as the target of, e.g., a mapped-item when the representation is to be re-generated)
(2) - Like in clause 3; an instance of PLib class, i.e.,
a - one class_BSU
b - a set of couple (property_BSU, value)
(3) - an association between the externally-defined representation (1) and the PLib class instance (2) that characterizes the fact that Plib class instance IDENTIFIES the externally-defined representation.

Note that the association between the class instance (2) and the externally-defined representation (1) means that the representation was (and is to be) generated by a particular class (called in Plib: functional

model class). The part/product associated with this representation also may, or not, be associated with a class, but it would be a different class.

Example: a pump instance is:

- identified as an instance of apump class with an (unique) identification property "supplier_id" = 123XX55
- represented by a representation generated by the "torque_pump_geom" class, with property "geometry_level" = "solid", "detail_level" = "standard" and "size" = 15 (simple example where the geometry of the pump depends only on one dimensional property).

Proposal **Date:** 23-Jan-98 **Proposer:** G. Pierra (ENSMA),

The three first requirements are identical to the one expressed by AP 212, and all four requirements will appear in AP 214 and should appear in AP 221. During the Florence meeting a joint proposal from AP 212 and AP 214 for the statement of these requirements and for their interpretation using the STEP IRs was presented by Guenter Staub during the joint WG2/WG10 meeting (ISO TC184/SC4/WG2 N345). There was a consensus on the proposed approach.

During the same meeting, this proposal was discussed within the SCR QC (ISO TC184/SC4/QC N039). There was an agreement to update the material from Guenter Staub, and to incorporate the result into the Interpretation Guidelines Document.

It is suggested to interpret this requirement exactly the same way as AP 212/214 to contribute to AP interoperability.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Changed from open to rejected. Requests additional things we need to put in the ARM to address externally defined properties. This appears to be a big change to the AP at this time. We do support the referencing of different external libraries.

Reject the issue. No additional capability (defined property or externally defined rep items)

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 3 **Alt. Numbers:** JAPAN-227-DIS-001 **Issue Date:** 19-Feb-98
Issue Title: Circular_ellipsoid Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

Mapping of the circular_ellipsoid is not given in the table.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Map it to the ellipsoid in Part42/v2.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Issue is wrong; it is in the mapping table. Agree with proposed resolution.

Comment **Date:** 03-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

Revise mapping table 5 to change the AIM element for circular_ellipsoid from revolved_area_solid to ellipsoid.

Issue Number: 4 **Alt. Numbers:** JAPAN-227-DIS-002 **Issue Date:** 19-Feb-98
Issue Title: Eccentric_cone Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

Mapping of the eccentric_cone to hybrid_shape is unncessary and incomplete.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Use instead the eccentric_cone now defined in Part42/v2.

Comment **Date:** 03-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve/Mitch

Revise mapping table 5 to change the AIM element from (shell_based_wireframe_model) to eccentric_cone and delete the reference path mapping.

Issue Number: 5 **Alt. Numbers:** JAPAN-227-DIS-003 **Issue Date:** 19-Feb-98
Issue Title: Eccentric_cylinder Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

Mapping of the eccentric_cylinder to hybrid_shape is unncessary and incomplete.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Use instead the eccentric_cone now defined in Part42/v2. It can model the eccentric_cylinder in this table as well.

Comment **Date:** 03-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve/Mitch

Need a mapping rule for the mapping of cylinder that the value of the ratio attribute equals one for an eccentric_cylinder.

Revise mapping table 5 to change the AIM element from (shell_based_wireframe_model) to eccentric_cone and delete the reference path mapping.

Issue Number: 6 **Alt. Numbers:** JAPAN-227-DIS-004 **Issue Date:** 19-Feb-98
Issue Title: Use of Part 42/v2 CSG Primitives

Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 4.2.55
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:** ARM

Description:

Choice of the csg_element is not effective with regard to part42/v2.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Match the choice to the selection in P42/v2. Use all the primitives given in Part42/v2, except_right_angular wedge. Only for square_to_round, a special primitive in this AP might be necessary. For this, mapping it to brep shape would be better.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

AI - MP - Ask for clarification as to whether more is needed than P42/V2. No longer needed based on resolution of other issues.

AI - SK - Definition of AO in 4.2.55 needs to be looked at. Should be consistent with the definition of CSG primitive in Part 42. Definition of csg_primitive from Part 42/V2 is "This select type defines the set of CSG primitives which may participate in boolean operations. The 3D CSG primitives are sphere, ellipsoid, right_circular_cone, eccentric_cone, right_circular_cylinder, torus, reducing_torus, block, faceted_primitive, rectangular_pyramid, and right_angular_wedge. The 2D CSG primitives which are all types of primitive_2d may participate in boolean operations with other two dimensional entities."

AI - ? - Look at square_to_round after we have reviewed and resolved other more general issues.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 7 **Alt. Numbers:** JAPAN-227-DIS-005 **Issue Date:** 19-Feb-98
Issue Title: Definition of Pyramid
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 4.2.197
Issue status: resolved **Classification** Editorial
Source: DIS Balloting **Subject:** AO Defs

Description:

Description of pyramid is inexact. Since it says "pyramid may be a trimmed pyramid" it may not have the apex and its side faces may not be triangular. Hence the description of the shape is incorrect.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Add a clause " if not truncated" .

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

Statement in 4.2.197 - Trimmed_pyramid statement needs to be removed.

AI - SK - Delete the last two sentences that refer to an eccentric_pyramid and trimmed_pyramid from the definition of pyramid. Eccentric_pyramid and trimmed_pyramid are no longer modeled as types of pyramid.

Issue Number: 8 **Alt. Numbers:** JAPAN-227-DIS-006 **Issue Date:** 19-Feb-98
Issue Title: Locating Mobile Equipment Within the Plant
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 4.1
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:** ARM

Description:

Mobil mechanical equipment, such as indoor crane, fuel rod exchanger, cannot be represented. The indoor crane is oftenly installed within nuclear reactor building in order to load and unload material from and to the reactor vessel.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Add support_mechanical_equipment UoF.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Covered but not very well by the current ARM as a plant_item. Reserved_volume covers its path and it can be associated with a plant_item. How is it located in the plant if it doesn't have a fixed location? Perhaps use its location where it is stored as its location. Covered by user_defined_xxx attributes.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 9 **Alt. Numbers:** JAPAN-227-DIS-007 **Issue Date:** 19-Feb-98
Issue Title: Eccentric_pyramid Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

Mapping of the eccentric_pyramid is incomplete.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Use instead a hexa_hedron or a rectangular_pyramid that are now available in Part42/v2.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

It is not a type of pyramid as stated in 4.2.64. MG proposed a generalized solution for defining CSG primitive requirements that are not addressed by Part 42/V2 (eccentric_pyramid, square_to_round, etc.) called plant_design_csg_primitive. At this point in time, this is not envisioned as a generalized csg shape requirement in the ARM. (Can be general or specific.) NS was concerned with the amount of overhead associated with implementing MG's proposal.

Issue - Should we use faceted CSG in AP 227?

AI - SK - Fix text in clause 4.2 to indicate that all primitives are types of csg_primitive.

Possible CSG solutions:
· Request further changes to Part 42.
· Use derived form faceted_primitive in Part 42/V2. Could be used for trimmed_pyramid.
· MG's proposal - plant_csg_primitive. Could be used for trimmed_sphere, hemisphere, etc.

CSG primitives with problems:
· trimmed_sphere
· hemisphere
· trimmed_block (Should it be called this since the only cases people are aware of only cut the block at right angles to a face of the block?) without cutting plane perpendicular to a plane of the block
· sloped_bottomed_cylinder (doesn't appear that this one is used too much - do not include in AP?)

AI - MP - Send out a request to folks requesting them to respond back as to whether the primitive sloped_bottomed_cylinder needs to be in AP within a month. Also check DuPont need for this.

· trimmed_cylinder (same as sloped_bottomed_cylinder if ends need to be sloped - issue above also applies)
· tube (only have Boolean result as a poor solution at this time)
· trimmed_pyramid
· eccentric_pyramid (need to define something to support it)

AI - MG - Need to define what the mapping should be for eccentric_pyramid.

· square_to_round

AI - SK - Clause 4.2 write-up for cone, trimmed_cone, etc. needs to be looked at. Looking for definitions that refer to incorrect subtype/supertype relationships.

AI - SK/MG - Change the definition of trimmed_cylinder to cover sloped_bottomed_cylinder? Use Boolean result with a specified name? Need to constrain the definition of trimmed_cylinder so that no cutting planes cut down the middle of the cylinder, i.e., parallel to the axis. Change definition to allow use of 2 cutting planes. If cutting planes are parallel, it should be mapped to eccentric_cylinder and cylinder. Put a name in .name "slope bottom cylinder". Also, planes do not intersect within the shape of the cylinder.

AI - MP - Question implementors forum on need for CSG construct for trimmed_block with corner cut off.

AI - MP - Submit Part 42 comments on trimmed_sphere and tube.

AI - MG - Remap trimmed_cone to eccentric_cone.

AI - MG - Remap trimmed_torus to reducing_torus.

NS - Part 42 will never really meet AP 227 requirements; the thinking is too different. Instead, use catalogue shapes (i.e., outside Part 42).

Square to round - This will be an AP 227 geometry type. Represent as a collection of cross sections? PP - In practice, this approach contains so much data as to be impractical. If you limit it to 2 cross sections (circle and a square), this may be OK. Call this approach Partially_defined_shape_representation_by_cross_section for now since it may be used for other shape representations (e.g., sloped_bottomed_cylinder, trimmed_sphere, hemisphere, trimmed_cylinder). This approach is similar to what MG sketched out earlier in the meeting

Issue - Is there an economical way to produce an elbow? Use of trimmed_torus would solve concern.

Issue - How do we address hollowness (i.e., a tube) with CSG? Need for wall thickness? Do we really have a need for a hollow pipe, but nothing else being hollow? A hollow pipe serves no purpose if other shapes are not hollow. This appears to eliminate the need for a tube. Consensus was that there is not a sufficient need for hollow components to justify keeping tube as an object in the ARM. Reference a catalogue/parametric based shaped requirements to cover hollowness, do not use CSG to address it. Also, piping size description provides outside/inside diameter and thickness.

AI - WB/SK - Remove the ARM object "tube" (and related mappings or references) from the AP.

Circular_ellipsoid - Need to constrain the Part 42 info with mapping rules.

AI - MG/NS - Address specific mechanism as to what approach to use to constrain the Part 42 information on circular_ellipsoid with mapping rules and put out to the team.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 10 **Alt. Numbers:** JAPAN-227-DIS-008 **Issue Date:** 19-Feb-98
Issue Title: Hemisphere Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:
Mapping of the hemisphere is incomplete, missing the second_operand.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama
Mapping it to the boolean_result of a sphere and a half_space created by a plane.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**
AI - MG/NS - What should the mapping for hemisphere be? MG/NS to develop solution.
See Issue No. 9 (JAPAN-227-DIS-007) for related problems.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 11 **Alt. Numbers:** JAPAN-227-DIS-009 **Issue Date:** 19-Feb-98
Issue Title: Extrusion Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:
Mapping of the extrusion cotains a grave mistake. composite_curve_on_surface cannot take as its composite_curve_segment.parent_curve, the trimmed_curve. composite_curve_on_surface can only take the following three, i.e., pcurve, surface_curve, composite_curve_on_surface. SEE WR2 of the composite_curve_on_surface. The mapping given in the table will crush the genuine P42-based CAD or viewing sytem because DERIVE get_basis_surface(SELF) malfunctions. This results in a fatal error.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Change all this mapping. I suggest, extruded_face_solid would be easier to handle, for its allows polyloop to define the face to be extruded.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

There is a problem with the mapping that needs to be fixed.

AI - MG - Fix the extrusion mapping problem.

Issue Number: 12 **Alt. Numbers:** JAPAN-227-DIS-010 **Issue Date:** 19-Feb-98
Issue Title: Solid_of_revolution Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

Mapping of the SOLID_OF_REVOLUTION cotains a grave mistake. composite_curve_on_surface cannot take as its composite_curve_segment.parent_curve, the trimmed_curve. composite_curve_on_surface can only take the following three, i.e., pcurve, surface_curve, composite_curve_on_surface. SEE WR2 of the composite_curve_on_surface. The mapping given in the table will crush the genuine P42-based CAD or viewing sytem because DERIVE get_basis_surface(SELF) malfunctions. This results in a fatal error.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Change all this mapping. I suggest, revolved_face_solid would be easier to handle, for its allows polyloop to define the face to be extruded.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Same as Issue No. 11 (JAPAN-227-DIS-009).

AI - MG - Fix the solid_of_revolution mapping problem.

Issue Number: 13 **Alt. Numbers:** JAPAN-227-DIS-011 **Issue Date:** 19-Feb-98
Issue Title: Trimmed_block Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

No mapping detail is given for the first operand of the Boolean for Trimmed_block.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Use half_space => plane to cut the shape.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Nikolay

Need resolution to issues discussed under Issue No. 9 (JAPAN-227-DIS-007).

AI - MG/NS - Discuss off-line. Depends on the resolution of Issue No. 9 issues.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 14 **Alt. Numbers:** JAPAN-227-DIS-012 **Issue Date:** 19-Feb-98
Issue Title: Trimmed_cone Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

No mapping detail is given for the first operand of the Boolean of Trimmed_cone.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Use half_space => plane to cut the CONE.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Nikolay

Need resolution to issues discussed under Issue No. 9 (JAPAN-227-DIS-007).

AI - MG/NS - Discuss off-line. Depends on the resolution of Issue No. 9 issues.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 15 **Alt. Numbers:** JAPAN-227-DIS-013a **Issue Date:** 19-Feb-98
Issue Title: Trimmed_cylinder Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

No mapping detail is given for the first operand of the Boolean for Trimmed_cylinder.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Use half_space => plane to cut the CYLINDER.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Nikolay

Need resolution to issues discussed under Issue No. 9 (JAPAN-227-DIS-007).

AI - MG/NS - Discuss off-line. Depends on the resolution of Issue No. 9 issues.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 16 **Alt. Numbers:** JAPAN-227-DIS-013b **Issue Date:** 19-Feb-98
Issue Title: Trimmed_pyramid Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan

Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:
No mapping detail is given for Trimmed_pyramid.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama
Use hexahedron now available in Part42/v2.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Nikolay
Need resolution to issues discussed under Issue No. 9 (JAPAN-227-DIS-007).

AI - MG/NS - Discuss off-line. Depends on the resolution of Issue No. 9 issues.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 17 **Alt. Numbers:** JAPAN-227-DIS-014 **Issue Date:** 19-Feb-98
Issue Title: Trimmed_sphere Mapping
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:
No mapping detail is given for Trimmed_sphere.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama
Cut it with a plane<=half_space_solid is better.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Nikolay
Need resolution to issues discussed under Issue No. 9 (JAPAN-227-DIS-007).

AI - MG/NS - Discuss off-line. Depends on the resolution of Issue No. 9 issues.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 18 **Alt. Numbers:** JAPAN-227-DIS-015 **Issue Date:** 19-Feb-98
Issue Title: Safety Classification
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 4
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:** UoFs
Description:
No consideration for safety classification, such as seismic safety requirements or regulations on earthquake-proof capability of equipments within a process plant. It is very much needed in process-plants along the pacific rim.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama
Create a safety classification UoF.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Panos/Jay
Change of scope. "Safety classification" is related to nuclear plants. ARM doesn't address any generic requirements for functional specifications such as safety, security, etc. (This lead to a more general discussion about the precise semantics that are intended to be conveyed with an AP versus a generic capability like PLIB or EPISTLE.)

What extensions do we have to add user-defined attributes to plant_items, plant_systems, etc.? Generic classification mechanism (once it is put in) should address this. What generic capability do we need to add? Add a generic capability for classification or use of value pairs? Addressed by user_defined attributes construct.

Issue - PP/JR - What attributes are optional vs required under a definition vs instance situation? Which attribute has priority if a definition and an instance define the same attribute?

AI - ? - Provide a summary of how the user-defined attributes will allow safety classification information to be captured.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 19 **Alt. Numbers:** JAPAN-227-DIS-016 **Issue Date:** 19-Feb-98
Issue Title: Consolidation of Shape ARM Elements
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.1 (Table 5)
Issue status: rejected **Classification** Minor Technical
Source: DIS Balloting **Subject:** ARM
Description:
Shape related ARM elements are scattered.

The same thing is aleady mentioned in Jay Roberts comments.

The illustration of mapped_item in Annex K.4, the figure is left-handed, and is very misleading to the readers.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama
Must consolidate the Shape related ARM elements.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to rejected per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Bill/Steve
Already have shape elements grouped under shape UoF. Reject the issue. Figure K.2 is left-handed, should be right-handed.

AI - WB - Fix figures. Done. Provided revised graphics for figure K.2 to SK 3/24/98.

Issue Number: 20 **Alt. Numbers:** JAPAN-227-DIS-017 **Issue Date:** 19-Feb-98
Issue Title: Radioactivity Classification
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 4
Issue status: resolved **Classification** Minor Technical

Source: DIS Balloting **Subject:** ARM

Description:

Radioacitivty related classification cannot be expressed by the ARM.

In Japanese nuclear related possibly hazardous facilities, special earthquake-proof capabilities are required for the architecture.

In the severest category, normally distinguished from others by zoning, the plant equipment is required to maintain its fonctionality even after a big earthquake of the scale of El Centro, Kanto, or Kobe earthquake. The second category requires that the architecutre maintains its shape, and contains the radioactivity and other poisonous material without leakage into the environment. The third is the normal archtectural earthquake proof ability to which applies the usual construction regulation, meaing it does not contain virtually any radioactive material within.

So it is basically a zoning, but note that all the major equipments and pipings contained within must hold agaist the possible natural disaster, including the "bombarding" by a jumbo jet crash. Depending upon the function of the component, required resilence is different. For example, LOCA(LOss of Coolant Accident of reactor vessel) is something that should never hapen in any case. So component related to this is much more severely looked upon about its safety in case of disaster.

I understand there is no such safety related requirement description within the AP. Moreover no way to relate this kind of safety to the Plant items, if I do not misunderstand badly.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Must include this in a "safety related classification" UoF

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

See Issue Nos. 8 and 18 (JAPAN-227-DIS-006 and JAPAN-227-DIS-015) for solution.

Issue Number: 21 **Alt. Numbers:** JAPAN-227-DIS-018 **Issue Date:** 19-Feb-98

Issue Title: Mapping of Shapes

Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: rejected **Classification** Major Technical

Source: DIS Balloting **Subject:** Mapping Table

Description:

Some of the vendors use solely on representation_item.name attribute to tell if the shape is mapped or not, instead of checking if it really has mapped_item.mapping_target, and mapping_source, mapping_source. Hence it results in an AP 227-specific implementation, lacking interoperability with shapes created in other APs.

Many of the attribute info are in fact unnecessary and/or redundant.

I think whether the 'in-situ' type of placement or the 'placed shape' type of placement is used must be judged, not upon these string contents, but whether the representation has a 'mapped_item' or it just directly has geometries with some placement (axis1_placement_2d. axis2_placement_3d, ...). So parser of translator must do a structural analysis.

Some of the traslators just use these string values to tell how it shoud be mapped or placed. I think it is dangerous; We cannot expect that the shape mapped with 'mapped_item' would be declared as such when it comes from other APs. Safe translation is to see if it has a mapped_item and related mapping source and mapping target axis2_placements. ¸ So how to decode information must be written in the AP.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

State creally by what the implemented program judge the mapping. It is not necessary to have 'placed_shape', neither 'in-situ' shape. It is sufficient to have the correctly connected entities concerned with the mapping. Strings are just an auxiliary information, in this case. Then, it must state so.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to rejected per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

Reject the issue. There are industry requirements for maintaining the specific name of the representation we have. The mapped_item is still used and may be checked and the name of the representation may be inferred.

AI - SK - Include MG's write-up on this subject in the response to this issue. (Note: Provided below.)

In-situ vs. Placed shape

Overview

There are two types of placement defined for plant_items within a building, plant or site. The types of placement are a result of the use of coordinate systems within the physical file. The coordinate system in which a plant_item may be defined is dependent on the way that a user specifies a coordinate system to define its shape. The shape of a plant_item may be defined in its own local coordinate system, or it may be defined in the coordinate system of a building, plant, or site. There are several implications on the use of these two diffferent methods of specifying the coordinate system for a plant_item that affect the way the plant_item is defined. These implications will be discussed before the details of the positioning are described.

Definining plant_items

There are two ways that plant items are defined in AP 227. First, a typical plant_item, called a plant_item_definition in AP 227, may be defined. For example, a centrifugal_pump may be defined so that its shape representation and other characteristic data can be specifed once and reused many times. There may be many of this kind of pump manufactured and sold for use in a plant. Each use of the pump is asserted to have the same characteristics as the typical one. This typical plant_item may have physical occurrences, represented in AP 227 as a plant_-item_-occurrence or its subtype planned_physical_plant_item, specified in a file which may be positioned with respect to another plant item, within a building, plant or site. The use of the typical plant_item implies the fact that the typical is defined in a local coordinate system and is placed into the coordinate system of a building, plant, or site, or with respect to another plant_item. An AP 227 file may have plant_item_occurrences and/or planned_-physical_-plant_-items that are each defined in their own coordinate system, or are defined in the coordinate system of a building, plant or site.

Defining coordinate systems

Coordinate systems in STEP are represented by an instance of geometric_representation_context. The coordinate_space_dimension attribute defines the number of dimensions that are in the coordinate system. The value of this attribute will be 2 for a two dimensional coordinate system or 3 for a three dimensional coordinate system. Each instance of geometric_representation_context represents a different coordinate system. In practice, when many different coordinate systems are defined, they are

each referred to as a local coordinate system. With each instance of geometric_-representation_-context an origin and orientation are implicitly defined to exist for the coordinate system that is represented by the geometric_representation_context. For example, a three-dimensional coordinate system, specified by a geometric_representation_context with a coordinate_-space_-dimension of 3, would have asserted an origin point at (0.0, 0.0, 0.0), x-axis orientation direction of (1.0, 0.0, 0.0), y-axis orientation direction of (0.0, 1.0, 0.0) and z-axis orientation direction of (0.0, 0.0, 1.0). All geometric entities that are geometrically founded (see 3.1.23 of ISO 10303-42 for the definition of geometrically founded) in that coordinate system are with respect to the asserted origin and orientation.

Any shape_representation can be defined in its own local coordinate system or in a shared coordinate system. The context_identifier attribute of geometric_representation_context should be used to identify the coordinate system, for example - 'site xxx coodinate system'. If the context_-of_-items attribute of the shape_representation of a plant_item references a geometric_-representation_-context which is referenced by the context_-of_-items attribute of another shape_representation, then the two collections of representation_-items (most likely geometric_representation_items) are being defined in the same coordinate system. If the shape_representation of a plant_item has a context_of_items that references a geometric_representation_context that is not shared by another representation then that context defines a local coordinate system for the plant_item.

Placed shapes

Figure 1 - Placed shape

Placed shapes are shapes of plant_items that are defined in their own local coordinate system and are placed into a target coordinate system through the use of the mapped_item entity. The mapped_-item entity defines the spatial relationship between the two geometric_-representation_-contexts. Figure 1 is an instance graph that depicts the instantiation of AIM entities for the definition of a placed shape of a pipe within the shape of a site for a plant. In the figure, the pipe is defined in its own local coordinate system through a physical definition. There is a physical instance specifed that has a shape that reuses the shape_representation defined for the pipe's physical definition. The site also has a shape_representation that is defined within its own local coordinate system. The shape_representation of the site contains a mapped_item that maps the shape_representation of the pipe into the shape_representaiton of the site. The two placment instances define the transformation from the piping coordinate system to the site coordinate system. The placement instances may but need not be in the items set of either representation.

In-situ shapes

In-situ shapes are shapes of plant_items that are defined in the coordinate system of the building, plant, or site in which they are located. Figure 2 depicts an in-situ shape of a pipe within the coordinate system of a site. In this case, there are physical instances of both the site and the pipe. Each physical instance has a corresponding shape_representation that collects those representation_--items that comprise its shape. In the in-situ case, the shape_representation of the

Figure 2 - in-situ shape

pipe references the geomteric_representation_context called 'site local coordinate system'. All of the geometry defined in the shape_representation of the pipe in this case is defined with respect to the origin of the site.

Issue Number: 22 **Alt. Numbers:** JAPAN-227-DIS-019 **Issue Date:** 19-Feb-98
Issue Title: PLIB Interface Description
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.2
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:**
Description:

Collaboration with PLIB is insufficient and scarce. Need more detailed interface description. For example relationship between the BSU code and the product_definition.id and product.id in AP 227.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Must include how to incorporate the PLIB defined shapes. Part20 of the PLIB is enough generic, not only for FORTRAN as it is often misunderstood.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch

Ensuring inclusion of the accepted PLIB interface in AP 227.

AI - MG - Need to look at PLIB from AP 214 and see how they defined the relationship between the BSU codes and the attributes of the externally_defined_xxx.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 23 **Alt. Numbers:** JAPAN-227-DIS-020 **Issue Date:** 19-Feb-98
Issue Title: PLIB Interface Description
Issue Owner: Hioshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):** 5.2
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:**
Description:

Collaboration with PLIB is insufficient and scarce. Need more detailed interface description.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Must include how to incorporate the PLIB defined shapes. Part20 of the PLIB is enough generic, not only for FORTRAN as is often misundersood.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch

Ensuring inclusion of the accepted PLIB interface in AP 227.

AI - MG - Need to look at PLIB from AP 214 and see how they defined the relationship between the BSU codes and the attributes of the externally_defined_xxx.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 24 **Alt. Numbers:** JAPAN-227-DIS-021 **Issue Date:** 19-Feb-98
Issue Title: AP 221 Interface Description
Issue Owner: Hiroshi Murayama - Toshiba **Country:** Japan
Doc number SC4 N580 **Clause(s):**
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:**
Description:

Collaboration with AP 221 is insufficient and scarce. Need more detailed interface description.

Proposal **Date:** 19-Feb-98 **Proposer:** Hiroshi Murayama

Must include how to interoperate the two APs. It seems that AP 227 overly define what is expected to be defined in AP 221; plant item topologies. Definition of them within AP 227 will make things complicated. Spatial configuration would better exclude topological connection of plant items.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Interface conformance class is being defined for AP 221. Will continue to examine as permitted by the time allowed.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 25 **Alt. Numbers:** JAPAN-227-DIS-022 **Issue Date:** 19-Feb-98
Issue Title: Detailed HVAC/Electrical Tray Design
Issue Owner: Kenji Araki - Hitachi **Country:** Japan
Doc number SC4 N580 **Clause(s):** 1
Issue status: resolved **Classification**
Source: DIS Balloting **Subject:** Scope

Description:

AP 227 does not cover any detailed data of ducting nor tray. In the nuclear and fossil power plant ducting and tray data need all of data how to fit and how to connect. I am not able to refer to a chemical plant, because I have no experience to work for any logical design and any detailed (3D layout) design of it. AP 227 has almost full detailed data which the 3D layout designer needs to make the piping system. I am talking about the 'fitting', 'valve', 'gasket' and other entites of children of 'piping component' entitiy. And I am also talking about the 'tee', 'elbow', 'coupling', and other entities of children of 'fitting' entitiy. Their entities are lacking on 'ducting component' and 'electrical_component'. We need to describe any methods about ducting fitting and cable tray fitting. We desire any additional entities about ducting and cable tray same as children of 'piping_component' entitiy. If AP 227 will not accept to install them itself, we can not do full data exchange about power plant 3D layout data.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Out of scope at this time to address connectivity of HVAC and tray. Detailed descriptions of HVAC, electrical, and structural component types are out of scope. The primary focus is the detailed descriptions of piping systems and the details at the plant item level for piping components.

WC - What is currently in the model is inadequate to get anything useful from it for HVAC modeling. They store descriptions of the component, not the interference envelopes. Results of discussion finally ended with the conclusion that the envelope shape covers the question of handling of interferences.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

AI - SK - Add the following clarification to the Scope. The connectivity and enumeration of non-piping systems (e.g., HVAC, electrical, and structural), while provided for by the AP 227 structure, is not the primary focus of the AP and is considered optional.

Issue Number: 26 **Alt. Numbers:** JAPAN-227-DIS-023 **Issue Date:** 19-Feb-98
Issue Title: Unique Component Ids
Issue Owner: Kenji Araki - Hitachi **Country:** Japan
Doc number SC4 N580 **Clause(s):**
Issue status: open **Classification**
Source: DIS Balloting **Subject:**

Description:

Another issue is about the catalogue data. I hope to be picked up that issue as a big topics for solving by all implementors. Our 3D layout CAD system has a unique ID or code number of plant parts. They are under JIS (Japan Industry Standard) code except the special nuclear parts. My overview is that AP 227 is just a standard of box into which we can input the process plant data. For example, we know structural_component.type' entity. However we do not know to put the exact name of data into the 'structural_component.type'. I can say 'wall', 'floor', and others. I believe we should deside the common name of that. That is an another issue.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mark/Bill

This is another example of generic classification. This is a part number for procurement purposes. Identification to a specific plant_item using a part number designation.

AI - MP - Get clarification from Araki on code/unique ID for plant parts.

AI - WB - Use of identifiers at the ARM level needs to be looked at.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 27 **Alt. Numbers:** KOREA-227-DIS-001 **Issue Date:**
Issue Title:
Issue Owner: **Country:** Korea
Doc number SC4 N580 **Clause(s):**
Issue status: resolved **Classification**
Source: DIS Balloting **Subject:**

Description:

The content of the documents is so complicated that the end users hardly understand in detail. It is also hard to recognize how to relate each clause and sub-clause. Although there have been plenty of effort to develop the STEP tools, no official or recommended tool has been announced so far.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

Expanding the use of the data model in clause 4. Revising the introduction to provide guidance on how different readers should use the AP document. Today there are no approved tools to use. Project will continue to investigate the use of any tools.

AI - ? - Revise the Introduction to provide guidance on how different readers should use the AP. Expand the use of the data model (figure 1) in clause 4.

Issue Number: 28 **Alt. Numbers:** UK-227-DIS-001 **Issue Date:** 04-Feb-98
Issue Title: Use of Group Construct for all Class/Classification Solutions
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.1, 5.2

Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** AIM

Description:

In common with other APs, AP 227 uses product_category, product_related_product_category, and product_category_relationship for the classification of 'products', but uses subtypes of group, group_assignment, and group_relationshop for classification of everything that isn't a product. This use of different constructs for the same concept is confusing and may lead to unnecessary overheads in implementations.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

The use of product_category etc. should be removed, and group used as the basis for all class/classification solutions. This should be implemented consistently across (at least) AP 221 and AP 231 as well as AP 227.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Product_category issue. Replace it with a classification assignment (group). Agree with the proposed resolution.

AI - MG - Identify and modify all mappings that make use of product_category, product_related_product_category, and product_category_relationship to use group, group_assignment, and group_relationship.

Issue Number: 29 **Alt. Numbers:** UK-227-DIS-002 **Issue Date:** 04-Feb-98
Issue Title: Naming of SELECT Types
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** AIM

Description:

In the AP 227 AIM several of the SELECT types that are used in the completion of management resource constructs have "plant_spatial_configuration_" as a prefix to the type name. This disables interoperability with AP 221 (and probably other relevant APs) have equivalent SELECT types with different names.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

AP 221, AP 227 and AP 231 should agree on common names for shared SELECT types.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Hooking up the management resource construct. Agree with issue. Use the xxx constructs.

AI - MG - Identify SELECT types that are shared across APs and develop common names. Use the "applied_xxx_assignment" practice.

Issue Number: 30 **Alt. Numbers:** UK-227-DIS-003 **Issue Date:** 04-Feb-98
Issue Title: Use of Weight vs Mass
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2.3.1.49
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** AIM

Description:

This entity type seems to treat mass and weight as synonyms; and it is not clear from the EXPRESS constraints whether a mass measure or a force measure is required within the representation.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Clarify requirements. If the requirement is for mass, then the term weight should not be used, and vice versa. If it is desired to note that in common engineering usage one of the terms may be used to refer to the other, then this should be in a NOTE. If the intent is that this representation may include representation_items for both mass and weight this should be stated, and the EXPRESS modified to state explicitly the requirements for mass measures and weight measures.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch/Steve

Referenced clause references mass and weight which makes it ambiguous. We should use a derived unit and remove mass. Search for any other "weight" and remove all references to mass.

AI - MG - Change the mapping of plant_item_weight to remove mass from both the mapping table and the plant_item_weight entity in the AIM (5.2.3.1.49). Also, add a mapping rule that the value uses a derived_unit.

AI - SK - Identify any other "weight" related objects and remove any references to "mass".

Issue Number: 31 **Alt. Numbers:** UK-227-DIS-004 **Issue Date:** 04-Feb-98
Issue Title: Numbering of Single Attributes in Clause 4.2
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 4.2
Issue status: open **Classification** Editorial
Source: DIS Balloting **Subject:** AO Defs

Description:

Application objects that have only one attribute do not define this as a separate subclause. This seems to be a ludicrous case of the ISO Directives and STEP Supplementary Directives triumphing over common sense. By including an attribute definition in a numbered subclause, it can then be referenced from another document.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Document all attributes (irrespective of how many there are for a given application object) in numbered subclauses. If SDs etc. prohibit this, change the SDs!

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Steve

Agree with recommendation, but this needs to be reviewed by the Quality Committee.

AI - SK - Pose issue to Jesse Crusey and document QC response to this issue.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 32 **Alt. Numbers:** UK-227-DIS-005 **Issue Date:** 04-Feb-98
Issue Title: Plant Coordinate System Constraint
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 4.2.170, 5.2.3.1.44
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:**
Description:
The Plant application object (and the plant entity data type in the AIM) require the presence of a coordinate system/origin for the plant. This is a constraint that may apply within AP 227 when a 3D model is present; however, since Conformance Class 1 excludes geometry (and, we assume, is intended to provide core overlap with other APs) it is not appropriate for this to be a universal constraint in the AP.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West
Either remove the constraint entirely, or modify it as: IF there is one or more 3D model associated with the plant THEN it shall have a coordinate system/origin.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert
The is no need within CC1 for a coordinate system. It does not hurt to have it.

AI - MG - Remove the WR5 rule from plant and make the attribute optional. Add an informal proposition that says if any plant_items have shape then there shall be a coordinate system defined for the plant.

Issue Number: 33 **Alt. Numbers:** UK-227-DIS-006 **Issue Date:** 04-Feb-98
Issue Title: Trimmed_block Definition Correction
Issue Owner: Fowler **Country:** UK
Doc number SC4 N580 **Clause(s):** 4.2.252-4.2.257
Issue status: resolved **Classification** Editorial
Source: DIS Balloting **Subject:** AO Defs
Description:
Issues were raised in the UK comments against the CD of AP 227 that pointed out that a Trimmed_block is *not* a type of Block. According to the AP 227 issues log these issues, and the proposed resolutions, were accepted. However, although the ARM diagrams have been corrected the error has persisted in the normative text.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler
Correct definitions, e.g.: A Trimmed_block is a type of Csg_element (see ...) that is the result of cutting a Block with a plane and removing one of the resulting sections.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline
AI - SK - All the csg_element subtypes should be checked to see if the "type of" statement in the definition refers to the correct supertype.

Issue Number: 34 **Alt. Numbers:** UK-227-DIS-007 **Issue Date:** 04-Feb-98
Issue Title: Renaming of Blank_fitting_classification
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2.3.1.2 (and others)
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** AIM
Description:
The name of this entitty type (and equivalents) is misleading. This is a class (something not has members), not a classification (the association between the class and its members).

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West
Change the entity name to blank_fitting_class. (But see also Issue UK-227-DIS-008).

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert
Agree. There are other things in the AIM that need to be checked.

AI - MG - Review all xxx_classification entities and confirm whether they are class or classification. If they are class, then change the name to xxx_class.

Issue Number: 35 **Alt. Numbers:** UK-227-DIS-008 **Issue Date:** 04-Feb-98
Issue Title: Use of Class vs Classification
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** AIM
Description:
This AP uses hard-coded entity types for a limited set of standard types (classes) , of plant items, systems, etc. Within the scope of AP 227 exchanges this may not cause problems; however, interoperability with other APs would require that a common mechanism is used for class/classification of plant items, etc. When (in particular) interoperability or integration with AP 221 is required, data that will be held in these entity types in AP 227 will be in AP 221 as instances of more general types (especially product_definition) with classification assocations to standard data elements that are included in the AP 221 class library. This applies to the following AIM-defined subtypes: blank_fitting_classification, connection_motion_classification, connector_end_type_classification, ducting_system, elbow_fitting_classification, electrical_connector_classification, electrical_system, flange_fitting_classification, flange_fitting_neck_type_classification, inline_equipment, instrumentation_and_control_system, line_less_piping_system, pipe_classification, pipe_closure_fitting_classification, piping_component_class, piping_component_definition, piping_connector_classification, plant, pipe_line_segment_definition, process_capability, reducer_fitting_classification, spacer_fitting_classification, specialty_item_classification, structural_load_connector_classification, structural_system, swage_fitting_classification, system_classification, valve_classification.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Options: (1) Remove the hard-coded subtyping and make use of the AP 221 class library as the basis for classification of plant items (2) Provide a standard mapping (preferably in computer interpretable form, i.e., EXPRESS-X) that shows how an instance of electrical_system in AP 227 related to an instance of product_definition that is a facility and is classified as 'electrical system' in AP 221.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Include a general classification ability and capability but retain the precise constructs in the AIM. These classes will be reviewed per UK-227-DIS-007.

AI - MG/? - Include a discussion in the AP that addresses the three types of external classification:
· a general classification to a user specified class,
· external sources,
· and specific classification types in the ARM requirements.
Use an example from the AP 221 library as an external source later. The 3 types of classification need to be added to the technical discussions and clauses 4 and 5.

Issue Number: 36 **Alt. Numbers:** UK-227-DIS-009 **Issue Date:** 04-Feb-98
Issue Title: Connection/Connectivity Mapping
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

The mapping of connection and connectivity across various APs needs to be reviewed with a high degree of urgency. The following usage of resource entity types have been observed for connection: shape_aspect_relationship (AP 227), shape_aspect (AP 210), shape_aspect AND shape_aspect_relationship (AP 221), product_definition AND product_definition_relationship (AP 221), …

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Ensure consistent interpretation of requirements for connection and connectivity across related APs.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Mark/Wes

How should we deal with shape for a functional connector?

AI - MG - Review the mapping of connection (especially functional) for the rationale for the mappings (i.e., to determine why connection and connectivity mapping was done the way it is). Why was the overhead in the current mapping put in? Convey results of review of current requirements and mapping of connectivity portion of the model back to APs 221 and 230. Put this in the technical discussions.

AI - WC/MP - Document an ARM example of connectivity to use for assessing/improving commonality with AP 221 representation of connector/connection characteristics.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 37 **Alt. Numbers:** UK-227-DIS-010 **Issue Date:** 04-Feb-98
Issue Title: Use of Property_definition vs Representation_item
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2.3.1.74 and others
Issue status: resolved **Classification** Major Technical
Source: DIS Balloting **Subject:** AIM

Description:

The use of property_definition and representation for the identification of properties and their description is not consistent with other APs (particularly AP 221). This has been raised as a WG12 (SEDS?) issue by Pascal Huau (France) - there seems to be a lack of clarity/understanding of the circumstances where properties map to property_definition and when to named representation_items, and what the latter means if the property has no numerical or textual description.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Resolution to this issue may not require to change to AP 227 - if the rationale for using property_definition, representation and representation_item is documented and more widely understood it can be applied in those APs where interoperability with AP 227 is required.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch ?

AP 227 has no properties that don't have numerical or textual values.

AI - MG? - Describe how AP 227 uses property_definition and representation structures in a fundamental concepts write-up to close issue.

Issue Number: 38 **Alt. Numbers:** UK-227-DIS-011 **Issue Date:** 04-Feb-98
Issue Title: Catalogue Definition
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2.3.1.3
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:**

Description:

This entity type appears to represent the (paper) document that is the published form of a catalogue, rather than on the contents of the catalogue. It is the latter that is useful.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Modify definition, and consider whether a catalogue is really just an external source, not an external source AND a document.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Need a clear explanation of what we can do.

AI - ? - Write an explanation to address what we can do. Consolidate piping_design_csg and advanced_csg into plant_csg.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 39 **Alt. Numbers:** UK-227-DIS-012 **Issue Date:** 04-Feb-98

Issue Title: Change Management Capabilities
Issue Owner: Fowler,King,West **Country:** UK
Doc number SC4 N580 **Clause(s):** 1, 4.1
Issue status: open **Classification** Major Technical
Source: DIS Balloting **Subject:**

Description:

The change management capabilities of AP 227 appear limited to electronic handling of engineering change orders. The requirement to identify, manage and respond to changes in data sets does not appear to be satisfied or, if this is the intention, it is not clear how ECO and data reconciliation is differentiated. Even if the focus of AP 227 is on snapshot data exchange, it is vital that recipients of exchange files should be able to determine the relationships that exist between data in different files.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler,King,West

Clarify AP 227's capabilities and add necessary technical content to handle management of changed data as well as engineering change orders.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Steve/Bill/

We need to discuss what are the change management requirements that need to be met by this AP. AP 227 defines that something has changed, not the rationale for the change.

Talks about changes to the data set. Management of data relationships is out of scope.

Role of identifiers need to be incorporated into the technical discussion of changes. This issue references capability that is outside scope (i.e., identify same products undergoing change in different exchanges.)

AI - MG/SK - Rename change_delta to change_pair. This isn't really describing a delta.

AI - MP/MG/WB - Look at whether we can merge change_pair into change_item.

Related to US - 26.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 40 **Alt. Numbers:** UK-227-DIS-013 **Issue Date:** 04-Feb-98
Issue Title: Shape Representation Types
Issue Owner: Fowler **Country:** UK
Doc number SC4 N580 **Clause(s):** 5.2.3.1.1, 5.2.3.1.27, 5.2.3.1.42
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:**

Description:

In the UK issues against the CD the distinction between ¸ advanced_csg_shape_representation and piping_design_csg_shape_representation was questioned. In the CD issues log this was rejected, stating that this is an application/domain requirement. In the DIS we find that the distinction has been removed from the ARM but persists in the AIM.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler

Either include requirements for different types of shape representation in the ARM, or remove the distinction in the AIM.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Steve

Mapping tables are unclear as to whether advanced_csg or piping_design_csg is to be used. Break was to differentiate between CSG shapes that would be found in any system and those that are more specialized. Could not figure out from what is in the AP which is advanced_csg and piping_design_csg. This is no corresponding ARM structure that allows you to figure it out. What kinds of things go in one versus the other?

Issue - We need to be able to explicate the AIM (mapping table) requirements back into the ARM.

Anticipation of conformance classes and capabilities of CAD systems was what initially determined the breakup of CSG primitives into 2 UoFs. Advanced_csg name is consistent with AP 225. A lot of the shapes were unique to piping design. Definitions in 5.2 fail to convey information about the differentiation between the two types of primitives. Advanced_csg is a more constrained CSG as it is now in the AP.

MG - Want to have the same code in both APs, e.g., the function shared with AP 225. Concern: this kind of policy is not overt in AP design - not that it's a bad idea, it's just a covert design rule that nobody knows.

Types of geometry now are:
· advanced_csg
· piping_csg
· hybrid_shape_rep
· site_shape_rep

Issue - The utility of BREP in the conformance classes. A decision was made not to map to BREPs, but to the primitives in Part 42/V2. This approach will delay the release of the AP 227 FDIS until Part 42/V2 reaches the DIS level.

Issue - Should there be some intermediate level of implementers conformance to the conformance classes?

Issue - Parametric shape.

Issue - Boolean operations. Should we have limits on what is allowed (just unions)?

Issue - Negative volumes.

Issue - Economy of geometry structures.

AI - MG/SK - Collapse all the CSG stuff into one class called plant_csg. Hybrid_representation is the other type of shape grouping.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 41 **Alt. Numbers:** UK-227-DIS-014 **Issue Date:** 04-Feb-98
Issue Title: Computer Interpretation of Data Elements
Issue Owner: Fowler, King, West **Country:** UK
Doc number SC4 N580 **Clause(s):** 4.2.73, issues log
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** AO Defs

Description:

In the response to issue UK-227-CD-92 (issue 396 in the issues log) we were surprised by the note that the equipment_characteristics are intended to be human readable only. How much of the data in this AP is intended to be computer interpretable? The use of reference data (such as, but not restricted to, the AP 221 class library) allows data in name/value pairs to be computer interpretable: the name is that of an identified type (class) of property or characteristic, whose nature can be standardized across systems and the value elements therefore subject to appropriate computer processing.

Proposal **Date:** 04-Feb-98 **Proposer:** Fowler, King, West

Add to the scope statement and to clause 4.2.x as necessary statements of the intended use of data elements - human or computer readable.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

Nothing in the AP 227 model prevents multiple classifications of a class. If you have agreement on the content and the meaning of what is being passed then it can be computer interpretable.

The intent is to provide for class and classification via reference to any external source and a human interpretable text field for equipment_characteristics (not equipment data sheet information).

Issue Number: 42 **Alt. Numbers:** UK-227-DIS-015 **Issue Date:** 04-Feb-98

Issue Title: Ducting_component Definition Clarity

Issue Owner: West **Country:** UK

Doc number SC4 N580 **Clause(s):** 4.2.60

Issue status: closed **Classification** Editorial

Source: DIS Balloting **Subject:** AO Defs

Description:

In our comments on the CD, we identified ambiguity in the description of the role of a ducting component:

The phrase "gaseous or airborne particulate matter" could mean "gaseous matter, or airborne particulate matter" or "gaseous particulate matter, or airborne particulate matter".

We observe that the definition has been modified, but that the modification does not resolve the ambiguity!

If the former is meant, it is restrictive, it excludes particulate matter that is fluidised but not airborne.

Proposal **Date:** 04-Feb-98 **Proposer:** West

Clarity/correct definition.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to closed per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

Use "gaseous matter and airborne particulate matter" as is already in the AP.

Issue Number: 43 **Alt. Numbers:** USA-227-DIS-001 **Issue Date:** 29-Apr-97

Issue Title: Csg_element and Shape_parameter definitions

Issue Owner: Manoj Dharwadkar **Country:** USA

Doc number SC4 N580 **Clause(s):** 4.2.55, 4.2.211

Issue status: closed **Classification** Minor Technical

Source: DIS Balloting **Subject:** ARM

Description:

1. Clause 4.2.55 defines the "Csg_element" Application object. It states:

"Each Csg_element may be one of the following: a Block, a Circular_ellipsoid, a Cone, a Cylinder, an Extrusion, a Pyramid, a Solid of revolution, a Sphere, a Square_to_round, or a Torus."

But if we refer to Figure G.6 - ARM diagram 5 of 25, all csg entities are shown as sub-types of csg_element.

For example, from Clause 4.0 it appears that "TRIMMED_BLOCK" is a child of "BLOCK" and "BLOCK" is a child of "CSG_ELEMENT". But if we look at Figure G.6, it appears that both "TRIMMED_BLOCK" and "BLOCK" are children of "CSG_ELEMENT".

Do you see this apparent discrepancy? Are we making a mistake in interpreting the IDEF1X notation used in Annex G? Can you please provide us your input/comments?

2. Figure G.6 also shows "Shape_parameter" as a child of "Shape_representation" element. But, neither Clause 4.2.211 (Shape_parameter) nor Clause 4.2.213 (shape_representation_element) mentions "Shape_parameter" as a child of "Shape_representation_element".

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

1. What appears to have happened is that the ARM was revised between the CD and DIS versions and the text in 4.2 was not revised to be consistent.

2. The issue is correct that the supertype statements in clause 4.2 for geometry are incomplete/incorrect. Assuming the ARM diagram is correct, the following changes need to be made:

Add the following to the Shape_representation_element supertype statement (4.2.213):

Shape_parameter

Make the supertype statement in Curve (4.2.56) an incomplete categorization

Add the following to the Csg_element supertype statement (4.2.55):

Eccentric_cone
Eccentric_cylinder
Eccentric_pyramid
Hemisphere
Reducing_torus
Trimmed_block
Trimmed_cone
Trimmed_cylinder
Trimmed_pyramid
Trimmed_sphere
Trimmed_torus
Tube
Remove the supertype statement in Block (4.2.4).
Remove both supertype statement in Cone (4.2.48).
Remove all three supertype statements in Cylinder (4.2.57).
Remove both supertype statements in Pyramid (4.2.197).

Remove the supertype statement in Sphere (4.2.227).
Remove the supertype statement in Trimmed_sphere (4.2.256).

Also, some of the subtypes do not contain the correct wording as a subtype.

Redefine Shape_parameter (4.2.211) as type of Shape_representation_element
Redefine Eccentric_cone (4.2.62) as type of Csg_element (4.2.55)
Redefine Eccentric_cylinder (4.2.63) as type of Csg_element (4.2.55)
Redefine Eccentric_pyramid (4.2.64) as type of Csg_element (4.2.55)
Redefine Hemisphere (4.2.101) as type of Csg_element (4.2.55)
Redefine Reducing_torus (4.2.200) as type of Csg_element (4.2.55)
Redefine Trimmed_block (4.2.252) as type of Csg_element (4.2.55)
Redefine Trimmed_cone (4.2.253) as type of Csg_element (4.2.55)
Redefine Trimmed_cylinder (4.2.254) as type of Csg_element (4.2.55)
Redefine Trimmed_pyramid (4.2.255) as type of Csg_element (4.2.55)
Redefine Trimmed_sphere (4.2.256) as type of Csg_element (4.2.55)
Redefine Trimmed_torus (4.2.257) as type of Csg_element (4.2.55)
Redefine Tube (4.2.258) as type of Csg_element (4.2.55)

Comment **Date:** 21-Oct-97 **Commentor:** Steve Kline **Action:**
Revised AP to incorporate changes described in Mitch Gilbert's proposed solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of the proposed solution.

Resolution: **Res Date:** 9/23/97 **Impl Resp:** Steve Kline

Incorporate the proposed solution.

Issue Number: 44 **Alt. Numbers:** USA-227-DIS-002 **Issue Date:** 29-Apr-97
Issue Title: Mapping of Shape_parameter
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

The mapping of Shape Parameter (Table 10 - p.601) is to a measure_representation_item, but the reference path does not list measure_representation_item. The Shape_parameter attributes name and value do specify the AIM element measure_representation_item, and complete the rest of the mapping.

Proposal **Date:** 26-Sep-97 **Proposer:** Mitch Gilbert

The mapping rule in Shape_parameter should be changed. The following should be added as the first line:

measure_representation_item <=

Comment **Date:** 20-Oct-97 **Commentor:** Steve Kline **Action:**
Revised the AP to incorporate Mitch's proposed solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of the proposed solution.

Res Date: 9/26/97 **Impl Resp:** Steve Kline

Issue Number: 45 **Alt. Numbers:** USA-227-DIS-003 **Issue Date:** 30-Apr-97
Issue Title: Site_shape_representation supertype/subtype mapping
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:

Site_shape_representation is an abstract supertype, so only it's subtypes-faceted_surface_representation and point_and_line_representation-can be instantiated on and AIM to ARM mapping. However, there is no unique reference path-the mapping tables give identical reference paths for both. Our thinking is that we would have to examine an instance of site_representation in the AIM and determine whether the path for point_and_line_representation to survey_point or faceted_surface_representation to facet_trigon exists for this entity to make a determination of what type of ARM entity to construct. Is this true and are we guaranteed in the AIM that only one of these paths will exist for each site_representation? In addition, is this the intended way to distinguish between different ARM representations of AIM site_representations?

Proposal **Date:** 23-Sep-97 **Proposer:** Kathy Tan/Mitch Gil

The distinction between faceted_surface_representation and point_and_line_representation will be the relationship to facet_trigon, and survey_point respectively. In the Express, Where Rule 2 of site_representation (5.2.3.1.71) specifies that there should only be "exactly one connected_face_set or geometric_curve_set".

The determination of which ARM subtype to use is by whether one of the items is a connected_face_set or a geometric_curve_set. However, there is an inconsistency between WR2 of the site_representation AIM entity and the ARM. The ARM object site_shape_representation has a relationship to breakline (see p. 622). The mapping says that it goes through geometric_curve_set to the polyline. WR2 does not allow this for the ARM faceted_surface_representation case. The mapping of this assertion needs to be changed to show how to get there for an ARM faceted_surface_representation.

There should be a mapping rule here in the reference path that shows that the item of the site_representation is a connected_face_set for the faceted case and a geometric_curve_set for the point_and_line case. This should be a DIS ballot issue that is submitted and fixed.

The inconsistency is a different problem. The definition of site_representation (see p. 688) needs to be fixed to take the presence of a breakline in the representation into account, and the mapping needs to be fixed. Zero to many instances of poly_line with the name of 'breakline' should be added to the allowed items in the site_representation, WR2.

Comment **Date:** 25-Sep-97 **Commentor:** Implementers **Action:** Mitch Gilbert
AI - Mitch Gilbert to provide revision to AIM per his discussion. Make this a separate issue (breakline). Mapping table needs to be adjusted to fix problem identified in this issue.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

AI - MG - Revise the mapping tables and clause 4 as needed to incorporate the proposed resolution to the issue. Look at site_shape_representation mapping to see how improvements could be made to clarify the mapping. The documented fix for breakline nees to be incorporated.

Issue Number: 46 **Alt. Numbers:** USA-227-DIS-004 **Issue Date:** 30-Apr-97
Issue Title: Site_plant reference path
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

There is no reference path for sited_plant, and the AIM element given renders this too ambiguous for AIM to ARM or ARM to AIM mapping. Our thoughts are that perhaps it should perhaps read along the lines of:

site <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition => etc.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The reference path needs to reflect the correct use of mapped_item, that is, the representation of the sited_plant is the mapped_representation for a mapped_item in the representation of the site.

Also, the reference path for Sited_plant.plant_site_orientation contains an error. The three lines

property_definition
property_definition ->
characterized_definition

should be corrected as:

property_definition
property_definition.definition ->
characterized_definition

The mapping of the reference for sited_plant to site should go through property_definition_relationship.

Add a description of the conventions for populating attributes which don't have a mapping specified in the mapping table to Annex C (Implementation specific requirements).

Add a description of the conventions for populating non-standardized attribute values in the reference path. (e.g., name, description, id) in the Implementers guide.

Comment **Date:** 18-Jun-97 **Commentor:** Jay Roberts **Action:**

For sited_plant, functional_connector_occurrence_satisfaction and catalogue_item_substitute, why is the reference path in these cases not follow the same format as in the mapping of other attributes?

Comment **Date:** 25-Sep-97 **Commentor:** Implementers **Action:** SWK/Implemente

A site_plant is modeling a role of another entity. Roles are usually reserved for relationships, so the modeling is somewhat screwy. A mapping rule should be added to the mapping of ARM Sited_plant to say that it is the definition attribute of a site entity as the assertion site to sited_plant indicates.

New issue for sited_plant. Mapped_item should be in the reference path for the plant, not the site.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:**

Hard to figure out how to instantiate a sited_plant. An example of how to do this would be useful. Current mapping seems to indicate that a site could only be associated with one plant, so you could only have one sited_plant for a site. Appears to be a disconnect between the ARM and how it was mapped in the AIM.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Bill/Jay/Mitch

AI - WB/JR - Add a write-up to annex C that discusses how to handle attributes that do not have a mapping specified in the mapping table either in an AIM element or by a mapping rule in the reference path. Start write-up using JR's listing of sources of values.

AI - MG - Incorporate the proposed solution.

Issue Number: 47 **Alt. Numbers:** USA-227-DIS-005 **Issue Date:** 01-May-97
Issue Title: Plant_item_definition mapping
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

In the plant_item_characterization UOF mapping table 9, there may be a problem with the plant_item_definition mapping.

According to the table, the ARM plant_item_definition is supposed to map to the AIM elements product_definition or externally_defined_plant_item_definition. In the reference path, there is no parentheses around the externally_defined_plant_item_definition, which would mean that externally_defined_plant_item_definition is always used.

Suggested Correction:
(externally_defined_plant_item_definition <=
externally_defined_item)
product_definition

continue from 4th line of current reference path

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

Clarification is being pursued from the Mapping Table Guidelines owner on what the practice should be. We think it should be:

(product_definition)
(externally_defined_plant_item_definition <=
[product_definition]
[externally_defined_item])

and continue with the mapping rule.

Other places in the mapping table will be affected to do this consistently within the mapping table.

Comment **Date:** 25-Sep-97 **Commentor:** Implementers **Action:**
Our proposed solution is dependent on the resolution developed by the qualification committee.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch Gilbert
AI - MG - Pursue the proposed solution and evaluate it's correctness with QC.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 48 **Alt. Numbers:** USA-227-DIS-006 **Issue Date:** 08-May-97
Issue Title: Plant_item_weight.weight_state reference path
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Editorial
Source: DIS Balloting **Subject:** Mapping Table
Description:
In Table 9 (p. 564) of the AP227 mapping table, plant_item_weight.weight_state's reference path seems to be missing a close-parenthesis. I believe it goes at the very end of the reference path.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert
The last character of the reference path should be the closing parenthesis. Add it.

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:** Steve Kline
The assessment is correct. The last character of the reference path should be the closing parenthesis.

Comment **Date:** 20-Oct-97 **Commentor:** Steve Kline **Action:**
Revised AP to incorporate proposed solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/23/97 **Impl Resp:** Steve Kline
Editorial. Agree with proposed resolution.

AI - SK - Incorporate the proposed change to the mapping table.

Issue Number: 49 **Alt. Numbers:** USA-227-DIS-007 **Issue Date:** 08-May-97
Issue Title: Instrument's reference path
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Editorial
Source: DIS Balloting **Subject:** Mapping Table
Description:
In Table 9 (p. 531), instrument's reference path seems to be missing a close brace, }, and a close bracket,]. I'm not sure where they go, but probably at the end. If the close brace goes at the end, then it would surround the entire reference path, and might be unnecessary.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert
The last line of the reference path is missing because the path is longer than one page. The reference path should be split to two pages and the following line should be added:

group.name = `control loop']}

This line was cut offf when the document was converted to A4 size.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/23/97 **Impl Resp:** Steve Kline
Editorial. Agree with proposed resolution.

AI - SK - Incorporate the proposed change to the mapping table

---

Issue Number: 50 **Alt. Numbers:** USA-227-DIS-008 **Issue Date:** 08-May-97
Issue Title: Nipple's reference path
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table
Description:
In Table 6 (p. 341) of the Mapping table, Nipple's reference path has a small mistake. The first two lines show how piping_component_definition is a sub-type of product_definition. Then the third line jumps back to piping_component_definition.

I think it should read:
piping_component_definition
{ piping_component_definition
classification_item = piping_component_definition
....
}
{piping_component_definition =>
product_definition
product_definition.formation
...
}

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The placement of the mapping rules is strange. The suggested solution is not quite right because the first two lines must appear outside of any rule, therefore, they will be moved before the second mapping rules so the that the visibililty of the IR subtype is within the flow of the rule. Other places in the table should be checked also for consistency.

Revise the reference path for the application element NIPPLE on page 341:

Change the first two lines of the reference path from:

```
piping_component_definition <=
product_definition
```

to

```
piping_component_definition
```

Change the middle of the reference path from:

```
group.name = `nipple']}
{product_definition
product_definition.formation ->
```

to

```
group.name = `nipple']}
{piping_component_definition <=
product_definition
product_definition.formation ->
```

Comment **Date:** 17-Oct-97 **Commentor:** Steve Kline **Action:**
Incorporated changes discussed in proposed solution into the AP.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/23/97 **Impl Resp:** Steve Kline

Agree with proposed resolution.

AI - SK - Incorporate the proposed change to the mapping table.

Issue Number: 51 **Alt. Numbers:** USA-227-DIS-009 **Issue Date:** 08-May-97
Issue Title: Piping_system_component to Piping_size_description reference path
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:
In Table 6 (p. 385) of the Mapping table, piping_size_description lists:

```
...
{property-definition=>
product_definition_shape}
property_definition
...
```

in its reference path. When piping_system_component refers to piping_size_description (p. 544) the reference path only lists property_definition.

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum
Revise the middle of the reference path for application element piping_system_component to piping_size_description from:

```
property_definition.definition
property_definition <-
```

to

```
property_definition.definition
{property_definition =>
product_definition_shape}
property_definition <-
```

Comment **Date:** 20-Oct-97 **Commentor:** Steve Kline **Action:**
Revised the AP to incorporate the change described in the Proposed Solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/25/97 **Impl Resp:** Steve Kline

Mapping. Agree with proposed resolution.

AI - SK - Incorporate the proposed change to the mapping table.

Issue Number: 52 **Alt. Numbers:** USA-227-DIS-010 **Issue Date:** 08-May-97
Issue Title: Straight_pipe and Reducer end_to_end_length reference path
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification**
Source: DIS Balloting **Subject:** Mapping Table

Description:
In Table 6 (pps. 419-420) of the Mapping table, straight_pipe.end_to_end_length's reference_path has representation_item.name='end to end length' in definition #1b, but not in #1a. I believe this line should be added to the #1a definition. I believe a similar line should be added to the #1 reference path in reducer.end_to_end_length (p. 393).

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum

Appears to be an A4 conversion cut off problem. Add the following line to the end of the reference path for application element end_to_end_length on page 419 and 393:

representation_item.name = `end to end length'}

Comment **Date:** 20-Oct-97 **Commentor:** Steve Kline **Action:**
Revised the AP to incorporate the Proposed Solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/25/97 **Impl Resp:** Steve Kline

Mapping. Agree with proposed resolution.

AI - SK - Incorporate the proposed change to the mapping table.

Issue Number: 53 **Alt. Numbers:** USA-227-DIS-011 **Issue Date:** 14-May-97
Issue Title: Piping_system_line_segment_termination.line_end_location reference path correction
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Editorial
Source: DIS Balloting **Subject:** Mapping Table

Description:

In Table 7 (p. 482), piping_system_line_segment_termination.line_start_location, the second to last line of the reference path reads:

shape_aspect.name = 'line end location'

It should read:

shape_aspect.name = 'line start location'

(I confirmed this thought with Bill).

Proposal **Date:** 20-Oct-97 **Proposer:** Steve Kline

Incorporate the change identified by Kathy into the AP.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Res Date: 10/20/97 **Impl Resp:** Steve Kline

Issue Number: 54 **Alt. Numbers:** USA-227-DIS-012 **Issue Date:** 14-May-97
Issue Title: Piping_system_line_segment_termination attribute definitions vs reference paths inconsistency
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

In Table 7 (pps. 481-482), piping_system_line_segment_termination.line_end_location and line_start_location may have other problems. The ARM definition describes it as a cartesian_point with X,Y,Z coordinates and an indicator such as Center of pipe or Bottom of pipe, but the reference path only lists it as a point which just has an attribute name. Should there be anything else in this reference_path like a reference to a cartesian_point?

The subtypes of point are cartesian_point, point_on_curve, point_on_surface, point_replica, degenerate_pcurve. The question here is whether it is always simply an explicit XYZ location or the XYZ could be defined some other way (like a point_on_curve, point_on_surface, point_replica, etc.) The second part of the issue would be the shape_aspect.description. Since there is no explicit attribute to give this information, some logic must be used to determine where that would go. The "center of pipe" would be a shape_aspect, and the description attribute would say what the interpretation of the shape_aspect would be.

The reference point that is pointed to is based on the relationships (related to 4.2.167 definition description that indicates a "specified point"). Line start and line end location are not well enough defined in that the reference point that is being pointed to is not well defined.

The mapping table tends to leave you with dangling product_definitions. This needs to be discussed among the experts and some of the original participants of the AP. What product is being defined by some of the product_definitions?

Proposal **Date:** 16-Mar-98 **Proposer:** Workshop

AI - MG - Break this down into two attributes that are appropriately named and then do the mapping.

AI - WB - Fix the AP 227 ARM for line_start_location and line_end_location.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 55 **Alt. Numbers:** USA-227-DIS-013 **Issue Date:** 14-May-97
Issue Title: Clarification of Piping_size_description.piping_size_description_id mapping
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

Piping_size_description.piping_size_description_id (p. 387) maps to representation. Representation.name is already filled in through the reference path of piping_size_description. Where does the id go?

My interpretation is that it maps to the instance identifier (e.g., #11000 in the demo file) of representation rather than a real value. Since we invent the entity piping_size_description and the attribute psd_id doesn't correspond to a real world identifier, we don't need to fabricate and store a value for it.

This mapping should be clarified if either the Piping_size_description.piping_size_description_id is not needed, or is put somewhere else.

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum

The attribute Piping_size_description.piping_size_description_id is only there because the entity is independent and the modeler thought "it needed something". Remove piping_size_description.piping_size_description_id from the ARM and the associated mapping of the attribute. A key attribute is not needed since there is no natural identifier associated with the entity. It was originally put in because the entity was an independent entity. This id currently does not have a STEP construct that it can be mapped to so the information is lost, which is why it is considered a problem.

Need to determine whether there are any other ids that have a similar problem. Shouldn't we be consistent if we are going to delete the piping_size_description_id as noted above. Look at Annex K discussion of ids. Mitch Gilbert's opinion is that they should not be there if they do not reflect real world data or requirements since the ARM is a requirements model, not a data model. If we're going to delete the id discussed above, we should make sure we're not leaving in other ids with the same problem.

Comment **Date:** 20-Oct-97 **Commentor:** Steve Kline **Action:**

Incorporated deletion of AE piping_size_description_id as detailed in Proposed Solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

This issue brings up the role of Identifiers in the ARM. If there is a natural ID, then put it in the ARM. Should probably delete the piping_size_description_id attribute.

AI - SK - Delete the piping_size_description_id attribute from the AP as indicated in the proposed resolution.

AI - WB - Look through the AP for those with similar concerns to the piping_size_description_id attribute and identify them to SK for deletion from the AP.

Issue Number: 56 **Alt. Numbers:** USA-227-DIS-014 **Issue Date:** 15-May-97

Issue Title: Clarification of supertype mapping

Issue Owner: Jay Roberts **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: open **Classification** Minor Technical

Source: DIS Balloting **Subject:** Mapping Table

Description:

Our assumption with the mapping tables is that the AIM elements and reference paths indicated for an entity are a superset of the same for all the subtypes of that entity. For an AIM entity to map directly to such an ARM supertype, some aspect of it's reference path must fall outside the reference path envelope defined by the reference path's of the ARM entity's subtypes.

Recently, we have noted that the AIM entities for CSG_ELEMENT are BOOLEAN_RESULT and CSG_PRIMITIVE. However, there are a number of subtypes of CSG_ELEMENT whose AIM elements are not BOOLEAN_RESULT, CSG_PRIMITIVE, or subtypes of these AIM elements. Examples would include ECCENTRIC_CONE, REDUCING_TORUS, ECCENTRIC_CONE, ECCENTRIC_CYLINDER, TRIMMED_TORUS, SOLID_OF_REVOLUTION, CIRCULAR_ELLIPSOID, REVOLVED_AREA_SOLID, and EXTRUSION.

On the other hand, the reverse assumption that the supertype AIM element(s) and reference path(s) constitute elements that must be mapped to the supertype does not hold up for CSG_ELEMENT. This is because every possible case of CSG_PRIMITIVE in the AIM maps to a subtype of CSG_ELEMENT in the ARM.

We could use some clarification on this issue in particular and the intent of the mapping table info for supertypes in particular.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The solid_model and shell_based_wireframe_model entities need to be added to the "OR" in this mapping to cover the cases discussed in this issue. There would need to be a mapping rule that the solid_model would need to be stand alone or used as a boolean_operand.

There is another issue here, however, because a shell_based_wireframe_model cannot be used as a boolean_operand. We could add a new entity that is a subtype of both shell_based_wireframe_model and solid_model, but I don't think that would satisfy the constraints of something that was considered a "solid model" by Part 42. If the process plant CAD systems could handle this kind of thing (CSG models defined by shell_based_wireframe_models), then it should be implemented in 227.

Comment **Date:** 16-Oct-97 **Commentor:** Mitch Gilbert **Action:** Mitch Gilbert

No reference path is necessary in this mapping. The OR cases should simply be added.

AI - Mitch - We will probably need to go and check the mappings of the assertions for these OR cases, but lets do that while we're working on the mapping table after the ballot.

Comment **Date:** 16-Oct-97 **Commentor:** Steve Kline **Action:**

Mitch's proposed solution to this issue indicated that we should add the OR cases of solid_model and shell_based_wireframe_model to the AIM element column of ARM element Csg_element (see page 248 of the DIS AP). Is there also a reference path for either of these two new AIM elements that should be added? Other cases of shell_based_wireframe_model have a reference path. Should I use the same path for this case? What about solid_model? Does it have a reference path and if yes, what should it be?

Comment **Date:** 17-Oct-97 **Commentor:** Steve Kline **Action:**

Added Solid_model and Shell_based_wireframe_model as OR AIM elements to AP.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Bill

Make an abstract supertype or define the specific mapping?

AI - MG/WB - Look at all the categorizations and determine whether they should be complete or incomplete, and what should the mapping from these be?

Will fall out from what we plan to do on the geometry parts of the model.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 57 **Alt. Numbers:** USA-227-DIS-015 **Issue Date:** 12-May-97

Issue Title: AIM global rule discrepancy

Issue Owner: Jay Roberts **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.2.4.12

Issue status: closed **Classification** Editorial

Source: DIS Balloting **Subject:** AIM

Description:

In rule 5.2.4.12, product_context_discipline_type_constraint contains an apparent discrepancy. The statement of the rule is:

"Every PRODUCT_CONTEXT shall have a DISCIPLINE_TYPE of 'process plant'."

However, the EXPRESS specification is:

WRI: SIZEOF(QUERY(pc <* product_context | NOT (pc.name = 'process plant')))
= 0;

This seems to be enforcing the rule on the NAME attribute of PRODUCT_CONTEXT rather than the DISCIPLINE_TYPE attribute.

Any comments or insights on this?

Proposal **Date:** 12-May-97 **Proposer:** Mitch Gilbert

It appears to be a typographical error. "pc.name" should, in fact, be "pc.discipline_type".

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 5/12/97 **Impl Resp:** Steve Kline

Incorporated Mitch Gilbert's proposed change.

Issue Number: 58 **Alt. Numbers:** USA-227-DIS-016 **Issue Date:** 11-Apr-97

Issue Title: Mapping of pressure attributes

Issue Owner: Kathy Tan, Bill Burkett **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: resolved **Classification** Minor Technical

Source: DIS Balloting **Subject:** Mapping Table

Description:

We noticed a need for additional information in the mapping of the attribute operating_pressure in ARM entity service_operating_case. Operating_pressure maps to measure_with_unit's 2 attributes:

value_component and unit_component.

Value_component refers to type measure_value. Measure_value's types include length_measure, mass_measure, time_measure, but doesn't have anything specifically dealing with pressure which is derived from length, mass and time. Some measure_value types that might be appropriate are ratio_measure, parameter_value, context_dependent_measure, and count_measure. Bill believes that only ratio_measure is appropriate in this case.

In any case, we think that the type of measure_value needed for operating_pressure should be explicitly written into the mapping table.

Any changes to the mapping table may affect all cases of pressure in the mapping table. I believe this includes:

piping_connector_service_characteristic's attribute design_pressure
service_operating_case's attribute operating_pressure
pressure_class's attribute pressure_rating
piping_system_line_segment's attribute design_pressure
stream_design_case's attribute's pressure

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum

Make pressure a ratio_measure in the mapping table.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Make pressure a ratio_measure and a derived unit.

AI - MG - Modify the mapping tables where "pressure" attributes are described to make pressure a ratio_measure and add a mapping rule for a derived unit for all instances of pressure.

Issue Number: 59 **Alt. Numbers:** USA-227-DIS-017 **Issue Date:** 29-Apr-97

Issue Title: Flange connection materials

Issue Owner: Kenji Araki **Country:** USA

Doc number SC4 N580 **Clause(s):** 4.2.174.3

Issue status: resolved **Classification** Editorial

Source: DIS Balloting **Subject:** ARM

Description:

How does DIS ARM deal with bolts and nuts(for example) of flange? Connection Material (4.2.174.3) includes bolts and gaskets. Are nuts part of that definition?

Proposal **Date:** 05-May-97 **Proposer:** Steve Kline

The definition of connection_material indicates that this attribute can identify a specific plant item (e.g., gasket) used at a plant item connection or can indicate the specification(s) for the connection material. The example for this attribute definition talks about bolts and gaskets for a flanged connection. Other connection materials such as nuts and washers are also covered by this attribute. We will modify the example for this attribute to make this clearer.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline

Add a "such as" to the example.

AI - SK - Add a "such as" to the example in 4.2.174.3 to clarify that bolts and gaskets are examples of things that are part of a piping connection.

Issue Number: 60 **Alt. Numbers:** USA-227-DIS-018 **Issue Date:** 02-Mar-97

Issue Title: Incomplete mapping of geometric entities

Issue Owner: Hiroshi Murayama **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.2, Annex A

Issue status: rejected **Classification** Minor Technical

Source: DIS Balloting **Subject:** AIM

Description:

A part of the problem is attributable to the limited power of the expression of Part 42, while some others might be considered as the inherent problem of AP 227.

Problems attributable to Part 42 - The available set of CSG primitives is not good. Some addition is necessary; e.g., tetrahedron, convex polyhedron, ellipsoid, etc.

The addition of a tetrahedron or convex polyhedron could be alternative, since any polyhedron can be built by the union operations of, or just by the juxtaposition of two or more tetrahedrons, while the convex polyhedron is geometrically identical to tetrahedron when the number of constituent planes is four. However, the identification of two primitives given by the tetrahedron or convex polyhedron methods detailed below is not always easy due to numerical errors caused in the process of intersection algorithm. Therefore, the simultaneous adoption of the two is a much more practical solution.

A tetrahedron, the symplex in three-dimensional space, is a CSG primitive defined by four cartesian points. All of the four points should not lie on the same plane.

```
ENTITY tetrahedron
  SUBTYPE OF (geometric_representation_item);
  point1: cartesian_point;
  point2: cartesian_point;
  point3: cartesian_point;
 WHERE
        wr1: ((point3 - point1) x (point2 - point1)) … (        point4 - point1) > 0;
END_ENTITY;
```

A convex polyhedron is a solid polyhedron in three-dimensional space, of which any of the two adjacent faces do not intersect at an angle larger than 180 degrees.

```
ENTITY convex_polyhedron
  SUBTYPE OF (geometric_representation_item);
  plane_list: LIST[4:?] OF plane;
END_ENTITY;
```

An ellipsoid is a CSG primitive specified by three semi-radii; radius1, radius2, and radius3 along the local axis specified by position.

```
ENTITY ellipsoid
  SUBTYPE OF (geometric_representation_item);
  radius1: positive_length_measure;
  radius2: OPTIONAL positive_length_measure;
  radius3: OPTIONAL positive_length_measure;
  position: axis2_placement_3d;
END_ENTITY;
```

Problems attributable to AP 227 - Half space solid that exists in Part 42 is excluded from AP 227 AIM, hence it unnecessaryily limits the range of the geometries to be mapped into. Note that since the entity half_space_solid can possess "geometry.surface", repeated use of Boolean operation can produce the ultimate shape of most of the objects, if not efficient.

Comment **Date:** 16-Jun-97 **Commentor:** Steve Kline **Action:**

A discussion of the Part 42 portion of this issue was held between Mr. Murayama and Mr. Ray Goldt during the San Diego ISO conference (6/97) to address how the proposed changes could be accommodated in Part 42. Information on the resolution of this issue is to be provided by Mr. Murayama.

Comment **Date:** 16-Jun-97 **Commentor:** Steve Kline **Action:**

The problem attributable to AP 227 was fixed in the DIS version of the AP. The TYPE statement in AP 227 was modified to be consistent with Part 42.

Comment **Date:** 20-Jun-97 **Commentor:** Nikolay Shul **Action:**

I think ellipsoid is either already there in part 42 revision or is very close. We may reasonably expect Ray Goult to include these into the next revision for us.

Comments on the proposed entity definitions:

- tetrahedron and ellipsoid are ok.

- I am not sure I like convex_polyhedron definition. Basically, I am concerned about having to recompute intersection lines - all I can get from the file is unbounded planes. Also, what if I want to point to a particular vertice of that solid - I can't transmit it. All in all, I'd rather see faceted b-rep which is not a lot more complex, but a lot better defined.

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**

What is the status of these additions to the new edition of Part 42. If these requirements have been introduced and included then we are done. If they haven't been introduced they should be. If there is not time to address the issue in Florence then an implementor should be assigned the task of writing up an issue against the CD proposal.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to rejected per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

Reject the issue. Proposed primitives will be covered under what is discussed under the Japanese comment resolutions. Refer to appropriate Japanese comments.

Issue Number: 61 **Alt. Numbers:** USA-227-DIS-019 **Issue Date:** 29-May-97

Issue Title: Translation of shape primitives

Issue Owner: Kenji Araki **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: rejected **Classification**

Source: DIS Balloting **Subject:** Mapping Table

Description:

Some shape primitives of ARM csg_element will be translated in the AIM into those of b_rep element. Inversely AIM shape primitives of b_rep element will be translated in the ARM into those of b_rep element, which is not the same, comparing with the first style before translation.

Example 1

ECCENTRIC_CONE, ECCENTRIC_CYLINDER, REDUCING_TORUS (ARM element)
(ARM -> AIM translation)
|

v
manifold_solid_brep or shell_based_wireframe_model (AIM element)
(AIM -> ARM translation)
|
v
b_rep_element (ARM element)

Example 2

TRIMMED_TORUS, SOLID_OF_REVOLUTION, CIRCULAR_ELLIPSOID,
REVOLVED_AREA_SOLID (ARM element
(ARM -> AIM translation)
|
v
REVOLVED_AREA_SOLID (AIM element)
(AIM -> ARM translation)
|
v
REVOLVED_AREA_SOLID (ARM element)

Comment **Date:** 20-Jun-97 **Commentor:** Nikolay Shul **Action:**
Digest first, details (for these interested) below...

* Mapping from ARM CSG to AIM b-rep, has to go. It brings in too many problems.
* Most CAD systems do not retain CSG tree internally. So a boolean difference becomes just a shape. Reconstructing boolean difference from it will not, generally speaking, be possible. A possible solution: limit allowed operations on CSGs to union (which is how most systems operate anyway).
* Some systems have CSG primitives which are absent from AP227. Converting these to b-rep will lose data intelligence. Not clear what to do with that as we can't account for each CSG primitive in each system. One (questionable) way out is to use name attribure in shape_rep.
* It seems to me that we need more than one geometry conformance class, roughly along CSG/b-rep/wireframe lines.

We should not introduce these problems at ARM-to-AIM level, that much is clear. It follows that all ARM CSG primitives should map one-to-one into AIM.

Comment **Date:** 23-Jun-97 **Commentor:** Bill Burkett **Action:**
Please be advised that AP 227 requirements for csg primitives were sumitted to Ray Goult WG3/T1 as far back as the Grenoble SC4 meeting (October, 1995). He has reported to me (in follow up queries) that all of the primitives we submitted - with one exception - will be included in the next edition of Part 42. The requirements were the list of those primitives found in the 227 ARM.

The only exception is the square_to_round. Ray told me that the committee either could not agree on or could not figure out how to parameterize this primitive, so they did not address this requirement.

Although I suggest that we confirm this with Ray, I think much of the concern over "what-to-do" about the process plant csg primitives isn't necessary.

Comment **Date:** 23-Jun-97 **Commentor:** Mitch Gilbert **Action:**
[response to Nikolay Shulga email dated 6/23/97 11:26 am]

>- Process Plant CAD system are not famous for geometry-related expertise, excepting CATIA, of course. PDS, which covers about (I think) 90% of the process plant design CAD market, is MicroStation-based and does not use ACIS kernel we supply with MicroStation Modeler: it is strictly a surface-rep based package. So is Jacobus s/w, which (we hope) will shortly have the 95% of the market :-)
>

Is this surface-rep based package common in the industry, or is it particular to the PDS product. If you say it covers 90% of the market, then maybe AP 227 should consider inclusion of a surface based geometric model (for which there is currently an AIC defined for both geometrically and topoligically bounded surfaces), along with an accompanying conformance class. It appears that requirements can easily be added to the shape_representation_element_usage application object to cover this possibility.

>- Is there a problem in extending AP227 geometry to cover the primitives specific to Process Plant? Some of that stuff is IMO too specific to Part 42.
>

The AIM development rule is that any entity types (containing attributes, except the management resource assignment entities) shall be defined in an integrated resoure part. This rule is there so that all of the APs in STEP can have a consistent basis. If we can define a primitive using some constraints on its composition using boolean_result or something like that then the answer is yes. Otherwise, we have to work with the Part 42 folks to introduce the Process Plant requirements into that part.

>- The problem with identifying a primitive by name is that it will be impossible to reconstruct e.g eccentric_cone parameters (two radii, height, offset, etc.) from a b-rep representation.
>

I thought Ray told me in San Diego that the Geometry committee will be including the eccentric_cone in the next edition of Part 42 anyway, but maybe not by this name. If we are talking about some parametrically defined elements, there is a construct in the New Edition proposal of Part 43 that may allow us to define these primitives but I would have to look at the proposed Part 42 to see if we could use them.

Comment **Date:** 23-Jun-97 **Commentor:** Nikolay Shul **Action:**
[response to Mitch Gilbert email dated 23-jun-97 10:31 am]

Thanks for the response. Questions/comments:

- Process Plant CAD system are not famous for geometry-related expertise, excepting CATIA, of course. PDS, which covers about (I think) 90% of the process plant design CAD market, is MicroStation-based and does not use ACIS kernel we supply with MicroStation Modeler: it is strictly a surface-rep based package. So is Jacobus s/w, which (we hope) will shortly have the 95% of the market :-)

- Is there a problem in extending AP227 geometry to cover the primitives specific to Process Plant? Some of that stuff is IMO too specific to Part 42.

- The problem with identifying a primitive by name is that it will be impossible to reconstruct e.g eccentric_cone parameters (two radii, height, offset, etc.) from a b-rep representation.

Comment **Date:** 23-Jun-97 **Commentor:** Mitch Gilbert **Action:**
[response to 6/20/97 email from Nikolay)

I think that the reason for this mapping is to satisfy the requirement for that shape to be used as a primitive. The problem is that, to the best of my knowledge, there is no eccentric_cone in Part 42 as a primitive. The choices then become either define it as a brep so that the shape may be captured for use as a primitive, drop the requirement for that kind of primitive, or make a change to Part 42. The latter one is somewhat time consuming, although the timing is probably good for that right now. I now see that you have discussed this in your Problem 1. Have you discussed the need for these primitives with the committee responsible for Part 42?

Comment **Date:** 24-Jun-97 **Commentor:** Nikolay Shul **Action:**
[response to Bill Burkett email dated 6/23/97]

Ok. I am told that the latest AP227 mapping table maps a bunch of ARM CSGs to b-rep. This mapping is being used in AP227 implementation. Should we change it?

I have a few concerns wrt CSG->B-Rep mapping.
1) Mapping to b-rep is easy; mapping back to CSG primitive may be impossible.
2) Substantial percentage of process plant CAD systems do not use b-rep at all.

It seems that these concerns are resolved if we get rid of CSG->B-rep mapping and use draft Part 42 instead. It does have tetrahedron and ellipsoid, but I do not see convex_polyhedron. I suspect Ray Goult' response would be - use faceted b-rep or surface_based_representation.

Issue 600: Draft Part 42 does have reducing_torus; I don't see eccentric cone and cylinder.

I also have a concern wrt CSG boolean results. Basically, any system that does not keep CSG results internally, will not be able to recreate the shape as CSG. Consider 45 deg. elbow; its shape currently is ((torus SUBTRACT block) SUBTRACT block). Once this is resolved inside CAD system, it may be only be able to export a b-rep or surface model.

One solution is to restrict CSG operations to unions only.

So, as it stands right now, some of the mapping issues may be resolved by using draft Part 42 entities. Some others are still open. To resolve these, we need more input from process plant CAD vendors.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to rejected per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

Reject. We are not going to translate any CSG stuff to BREP.

Use of Part 42/V2 will resolve the mapping.

Issue Number: 62 **Alt. Numbers:** USA-227-DIS-020 **Issue Date:** 29-May-97
Issue Title: Additional geometric entity mappings
Issue Owner: Kenji Araki **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.2, Annex A
Issue status: OBE **Classification**
Source: DIS Balloting **Subject:** AIM

Description:
New AIM EXPRESS definition proposal for Part 42.

```
ENTITY OBLIQUE_CIRCULAR_CONE
  SUBTYPE OF (geometric_representation_item);
  position: axis1_placement;
  height: positive_length_measure;
  radius_start: length_measure;
  radius_end: length_measure;
  extruded_direction: direction;
END_ENTITY; -- OBLIQUE_CIRCULAR_CONE

ENTITY AIM_TORUS
  SUBTYPE OF (geometric_representation_item);
  position: axis1_placement;
  major_radius: positive_length_measure;
  minor_radius: positive_length_measure;
  angle: positive_plane_angle_measure;
END_ENTITY; -- AIM_TORUS

ENTITY CIRCULAR_ELLIPSOID
  SUBTYPE OF (geometric_representation_item);
  position: axis1_placement;
  radius: positive_length_measure;
END_ENTITY; -- CIRCULAR_ELLIPSOID

ENTITY REDUCING_TORUS
  SUBTYPE OF (geometric_representation_item);
  position: axis1_placement;
  major_radius: positive_length_measure;
  start_radius: positive_length_measure;
  end_radius: positive_length_measure;
  angle: positive_plane_angle_measure;
END_ENTITY; -- REDUCING_TORUS
```

Comment **Date:** 16-Jun-97 **Commentor:** Steve Kline **Action:**
A discussion of the Part 42 portion of this issue was held between Mr. Murayama and Mr. Ray Goldt during the San Diego ISO conference (6/97) to address how the proposed changes could be accommodated in Part 42. Information on the resolution of this issue is to be provided by Mr. Murayama.

Comment **Date:** 17-Jun-97 **Commentor:** Hiroshi Mura **Action:**

Well, on behalf of the chair of Part42, I'll explain some of the principles of Part42 revision. When we propose something to Part42, the definition must be "nearly" perfect, and must prove that it has universal ussage to be included into Part42(I know it is needless to explain to you, but I have to confirm this point), not only it has some value to some company or to some specific project. If not, it would be just scrapped as " non prima facie" proposal.

Because, as you well understand, Part42 is the "essense" of the STEP as a CAD standard, and many CAD vendors have already implemented this with huge amount of investment. Any change in the Part42 entails revision of almost all the AP's that use geometries.

In this sense, I understand very well the precaution of Mr. Ray Goult. More over it must be accompanied with the full text that explain well the context in which it is used and the meaning of each member of entity.

Unfortunately, most of the items of the proposal by our compatoriot did not satisfy this criterium. That's why Mr. Ray Goult rejected eventually all the proposal except one: that is just a special case of the "ellipsoid" that I proposed in Chester and is already in Part42-v2 WD.

Eccentric cone and partial torus were rejected because the first does not guarantee the correct shape and the second was geometrically wrong in definition.

Reducing torus was rejected because the definition was erroneous and it can be created as a kind of planar_swept_surface (and its sister enitty) that is proposed and in the draft already.

I think, first and foremost, the idea of ARM to AIM translator is wrong. That is to say, you can do it in one way but it is only natural that when you do the reverse AIM to ARM many of the informaiton would be lost. And I think, this misconcept does not justify any necessity to change the Part42. I myself would not buy that idea.

What I feel appropriate is to send in parallel by PLIB(external document) the original parameters with which an ARM csg object was created and from which a transformed Brep object in AIM is created. Probably it may necessitate a (minor) revision of AP227 itself.

I would like to hear opinions from wider cornes of the industry to go more forward, for the SEDS, if necessary. And I believe, too, that mathematical schema to be included into the Part42 has a big potential to be used to describe control and constraints that exist in many process-plant chemical processes and in operation. Mr. David Lead would be more able to explain what are the potential uses of this schema for Process industry.

Comment **Date:** 18-Jun-97 **Commentor:** Nikolay Shul **Action:**

Commentary due by Nikolay by 18-July-1997.

Comment **Date:** 20-Jun-97 **Commentor:** Nickolay Shu **Action:**

Some of the proposed primitives will have to stay AP227-specific; if these are needed, we should include them in the AP

My own issues...

*¸ Problem 1. The current set of CSG primitives available in Part 42 is not sufficient - most process plant design systems have a lot more primitives, mostly specialized for the use in this specific industry.

There are three possible solutions.

1) Add corresponding primitives to Part 42.
- That is not going to happen - at least not to the extent we need. We may be able to push through a few more or less generic primitives, but definitiely not all we need.

2) Add CSG primitives we need to AP227.
- That is an option; we do have to make sure the underlying geometry is valid. But - how do we know that it is an exhaustive set, i.e. 6 months down the road somebody else does not come up with yet another primitive.

3) Part 42 hierarchy allows to use any solid (eg, solid of revolution, b-rep, etc.) as a CSG primitive. We can model the primitives missing from AP 227 using these representations.
- That is also an option. However, recognizing the underlying CSG primitive from its B-rep representation will not, in general, be possible - the receiving system will have to treat it as an arbitrary shape. Thus losing some of the intelligence.

To me, a realistic solution will be somewhere between 2) and 3). I.e., we'll add some geometric primitives and use other solid representations for the remaining (hopefully small) set. The resulting loss of intelligence should not be that bad.

I also thought of using name attribute of shape_representation to keep that info. In the following example skewed_pyramid is represented by a set of trimmed surfaces

```
#10 = trimmed_surface(...);
#20 = trimmed_surface(...)
<...>
#1200 = shape_representation('skewed_pyramid_primitive', (#10, #20,
...));
```

I am not sure I like it. Ideas are welcome...

*¸ Problem 2. Receiving system gets a CSG representation - e.g.

```
#10= cone(...)
#20=block(...)
#30=boolean_result(#10, #20...)
```

The receiving system (which is the case for most of them) can't retain CSG expression - i.e. the fact that the shape it has was created by substracting #20 from #10. Thus again losing the intelligence.

In some cases, that is quite acceptable, in some it may be not acceptable at all.

Most current systems seem to get out of this by using unions only - i.e. it is always possible to

get to the original shapes. That may be a reasonable restriction. An alternative is having more geometry-related conformance classes.

Comment **Date:** 23-Jun-97 **Commentor:** Mitch Gilbert **Action:**
[response to 6/20/97 email from Nikolay]

>My own issues...
>
>* Problem 1. The current set of CSG primitives available in Part 42 is not sufficient - most process plant design systems have a lot more primitives, mostly specialized for the use in this specific industry.
>
>There are three possible solutions.
>1) Add corresponding primitives to Part 42.
>- That is not going to happen - at least not to the extent we need. We may be able to push through a few more or less generic primitives, but definitiely not all we need.
>

Can you define a small generic set of primitives that can be constrained to meet all of the needs of the process industry?

>2) Add CSG primitives we need to AP227.
>- That is an option; we do have to make sure the underlying geometry is valid. But - how do we know that it is an exhaustive set, i.e. 6 months down the road somebody else does not come up with yet another primitive.
>

If the mapping to a brep is insufficient, then these primitives would be impossible to map to the AIM if the primitives are not in Part 42.

>3) Part 42 hierarchy allows to use any solid (eg, solid of revolution, b-rep, etc.) as a CSG primitive. We can model the primitives missing from AP 227 using these representations.
>- That is also an option. However, recognizing the underlying CSG primitive from its B-rep representation will not, in general, be possible - the receiving system will have to treat it as an arbitrary shape. Thus losing some of the intelligence.
>

I think a good compromise would be to standardize the names for these primitives in the ARM, and then we could clue in a post-processor by constraining the representation_item.name for the manifold_solid_brep entity, for example.

>To me, a realistic solution will be somewhere between 2) and 3). I.e., we'll add some geometric primitives and use other solid representations for the remaining (hopefully small) set. The resulting loss of intelligence should not be that bad.
>
>I also thought of using name attribute of shape_representation to keep that info. In the following example skewed_pyramid is represented by a set of trimmed surfaces
>
>#10 = trimmed_surface(...);
>#20 = trimmed_surface(...)
><...>
>#1200 = shape_representation('skewed_pyramid_primitive', (#10, #20,
>...));
>
>I am not sure I like it. Ideas are welcome...
>
>*¸ Problem 2. Receiving system gets a CSG representation - e.g.
>#10= cone(...)

>#20=block(...)
>#30=boolean_result(#10, #20...)
>
>The receiving system (which is the case for most of them) can't retain CSG expression - i.e. the fact that the shape it has was created by substracting #20 from #10. Thus again losing the intelligence.
>
>In some cases, that is quite acceptable, in some it may be not acceptable at all.
>
>Most current systems seem to get out of this by using unions only - i.e. it is always possible to get to the original shapes. That may be a reasonable restriction. An alternative is having more geometry-related conformance classes.
>

All this sounds quite reasonable. I would suggest that you try to work with the Part 42 project team to get as many primitives as you can into the second edition of Part 42 either now, or through the CD ballot cycle.

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**
The suggestion from Nikolay that we can add some primitives in AP 227 as we need them is nice, but impossible using the STEP methodology, which ensures that all AIM constructs are derived from the vocabulary defined in the integrated resources. We cannot add entities with attributes to an AIM schema. We do, however, have a couple of options:

1. Work to get the primitives required by process plant CAD data exchange into Part 42. There is no rule that the only primitive allowed in Part 42 must be universal for MECHANICAL CAD systems (which is what Ray probably means when he says "universal"). If there are CSG primitives that are common to all process industry CAD systems, then an argument could be made to include them in Part 42 as long as the appropriate homework was done in advance, IMO.

2. Put the requirements into 227. The general problem with using Part 42 for our specific purposes is that the way it is modeled is heavily constrained. The result of the modeling methods is that Part 42 is not generic at all, but very strictly controlled. The specific problem this presents is the way to use these "primitives" in the CSG model. There are a couple of alternatives that could be persued along this line:

- We could use some "fudging" in the AIM and make a two level supertype/subtype hierarchy (which could be the basis for an AIC if that is useful), that has a process_plant_primitive as a subtype of solid_model and representation or shape_reprsentation, and each type of primitive as a parametrically defined entity below that. The problem with this is that it isn't really a solid_model, per se, and could be construed to be a "stretch" of the semantics of the solid model.

- Another thing we could do in this respect, is to have the parametrically defined primitives being "alternate" elements of representation for the brep, wireframe or other Part 42 solid models that are used in the csg_solid as a boolean_operand. The alternate relationship could then be specified using a representation_item_relationship entity. The issue here is that representation_item_relationship is not in the 1994 edition of Part 43, but is in the NWI/CD proposal. There is a timing issue there.

3. Pursue the development of a 100s series part for process plant CAD. This option would be very time consuming and resource intensive.

Comment **Date:** 24-Sep-97 **Commentor:** Implementers **Action:** MG/MD/NS
Long-term solution is to get what we need from the revised version of Part 42. The problem with this is the timing of getting the new version of Part 42 to a DIS level so we could reference it to support going to FDIS.

Another solution is to use faceted brep, etc. and give the geometry a name so people know what it is.

Nikolay's 24 September 1997 email identifies those new Part 42 constructs that we can use. We need to submit ballot comments to the new Part 42 for what is not there that we need.

What working practices do we need to use until we get Part 42 changes we can use?

Mitch Gilbert's proposal in the first bullet of Item 2 of his 23 September 1997 comment is a possible workaround. This issue needs to be carried forward as a DIS comment and the workaround put together during the comment resolution process.

AI - Each of the needed primitives and their attributes needs to be defined, so we can get them eventually into Part 42, and can use them in the interim solution. Nikolay is the lead with the other CAD vendors participating. Send an email to Nikolay with the schedule for doing this. Need to prepare comments on AP 227 DIS and also send similar comments to Part 42 CD when it comes out. Send copy to Murayama also.

Mitch Gilbert feels that these new primitives should not have their own coordinate system. This will ease any transition to Part 42. Orientation is a needed attribute of the new primitives. It will have an axis_placement, not a specific coordinate system for the primitive. New primitives should not be a subtype of shape_representation.

AI - Manoj to send Mitch the parameters for Pyramid and he will generate the EXPRESS for it, so it can used in the demo file. This will help us work out the bugs.

Comment **Date:** 24-Sep-97 **Commentor:** Nikolay Shul **Action:**

Quite a few CSG primitives in ARM got mapped into a boolean result in the AIM. Right now, that creates problems for the AIM<->ARM translator Jacobus is working on. That would create a similar problem for anybody trying to read a primitive e.g. trimmed pyramide from a boolean result in the data file. In general, such operation is not guaranteed to be successful and is in most cases quite complex. So the problems Jacobus is seeing in ARM<->AIM translator are really the same problems people would have translating to/from AIM-based file from/to their systems - if these systems operate in a way similar to ARM (as they should if the ARM is any good, right?)

Proposed: boolean_results in the mapping table to be replaced with primitives from the Part 42 -second edition. It incorporates most of the changes AP227 asked for some time ago. In particular, Jacobus is likely to use pyramid entity from Part42/2 in the demo as we have no other choice. Introduce a catch-all faceted-brep for cases that don't fit and either live with the resulting loss of intelligence or put a specified name into b-rep name (faceted_b_rep('square-to-round', (...); -- now I can try restoring square-to-round primitive out of this.. good luck). The same applies to primitive-to- b-rep mapping; I think we have some of these, too; the general principle is that we can't map a primitive to a boolean_result or b-rep in most cases as the reverse process is impossible or impractical or both.

Part 42/2 has reducing torus (part of the torus with uniformly changing minor radius), right_angular_wedge(a block cut by a plane perpendicular to one of the faces), rectangular_pyramid (classic regular pyramid), generalized_pyramid (truncated and/or skewed), tetrahedron, ellipsoid, half_space, box_domain, a bunch of 2-d primitives, swept_face/swept_area stuff that simplifies building complex extrusions, etc. Manoj has Part 42/2; we should be able to obtain an electronic copy from Ray Gould if need be.

The primitives affected:

Eccentric_Cone - mapped to hybrid_shape_rep - no primitive available, mape to surface model, raise the issue with Part 42 editor.

Eccentric_Cylinder - same as above

Eccentric_Pyramid - is apparently covered by Par42/2 generalized_pyramid. Currently mapped to boolean_result.

Hemisphere - currently mapped to boolean_result; no primitive in Part 42/2. Should be either a swept_area solid (with name set to 'hemisphere'?) or a primitive. Raise issue with Part 42 editor.

Pyramid - currently boolean_result. Part42/2 has rectangular_pyramid

Reducing_Torus - currently hybrid_shape_rep. Part42/2 has a primitive of the same name.

Square_To_Round - AFAIR, Part 42 team had a problem trying to figure out parametrization for this. Leave as hybrid_shape_rep (catch-all surface rep) for now.

Trimmed_Block - currently boolean_result. Part42/2 has right_angular_wedge that does the same thing.

Trimmed_Cone, Trimmed_Cylinder, Trimmed_Torus - currently boolean_result, I have no good solution for this.

Trimmed_Pyramid - currently boolean_result. Part42/2 generalized_pyramid covers this.

Tube - currently boolean_result. That may actually be acceptable to leave as boolean_result with specified name; but a primitive is a better option. Talk to Part42 editor.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to OBE per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

Is there a distinction between oblique and eccentric that needs to be modeled? OBE. All of the proposed entities were incorporated into Part 42 in some form.

Remove distinction between piping_csg and advanced_csg from AP. Call them all advanced_csg CSG elements. JR - Why do we even use this top level classification?

Use of Part 42/V2 will resolve the mapping.

Issue Number: 63 **Alt. Numbers:** USA-227-DIS-021 **Issue Date:** 18-Jun-97

Issue Title: Use and purpose of shape_parameter

Issue Owner: Nikolay Shulga **Country:** USA

Doc number SC4 N580 **Clause(s):** 4.2

Issue status: open **Classification** Minor Technical

Source: DIS Balloting **Subject:** AO Defs

Description:

1. The definition is not very useful. What is the intent of shape_parameter?
2. A shape parameter is a complex issue that needs to be investigated fully at the ARM level. Probably used to add additional parametric values.

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum

The shape_parameter was included in response to CD ballot comment which resulted in Issue No. 317. The purpose of shape_parameter is so that information about the shapes found in catalogs may be exchanged.

This object is used for specifying a specific value of a shape parameter. Add an example to the definition of shape_parameter (e.g., diameter and 5.6 corresponds to name and value of diameter attributes) in Clause 4.2. Add a note: A use of this structure is to provide a generic capability to reference classes of plant items by a dimensional characteristic (e.g., 2 inch pipe). Add a second note: It was not the intent of the object to use this structure to create a geometric representation of an item. The effective use of this structure requires an agreement between the exchanging parties as to the meanings of the names so that they can understand the information being exchanged.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Bill

Provide an example of how to use shape_parameter. Originally created for parameterized parts.

Check with NS. All we plan to do is an example of how to populate shape_parameter.

AI - WB - Take a cut at example an example of how to populate shape_parameter and then circulate.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 64 **Alt. Numbers:** USA-227-DIS-022 **Issue Date:** 18-Jun-97
Issue Title: Column Lines
Issue Owner: Kenji Araki **Country:** USA
Doc number SC4 N580 **Clause(s):** 4.2, 5.1, 5.2
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** AIM
Description:

Column lines represent grid and coordinate system used as a reference throughout the design phase. Nothing in AP 227 represents column lines.

Entity building
…
column_line_definition
…
end_entity; --- entity_building

ARM	AIM element	Reference_path
mapping column_line_definition	representation	plane=> elementary_surface=> surface=> geometric_representation_item=> representation_item<- representation.items[I] representation {representation.name='column line'}

Can ARM entity reference_geometry (4.2.201 - p. 112) be transferred or redefined to apply to a building as opposed to a plant_item? Are there other reference items needed for a building that must also be considered?

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum
Is column line one type of reference_geometry, like a centerline? Reference_geometry can be either a 2 or 3 dimensional shape. AP 225 had this issue raised against it and did something to fix it. Mapping using reference_geometry for column line should be similar to centerline. Fix in the ARM using reference_geometry.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Bill
Probably already have this under reference_geometry. Need to investigate.

AI - WB - See what AP 225 did to address column lines and compare to what we have in the ARM.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 65 **Alt. Numbers:** USA-227-DIS-023 **Issue Date:** 18-Jun-97
Issue Title: Plant North
Issue Owner: Kenji Araki **Country:** USA
Doc number SC4 N580 **Clause(s):** 4.2, 5.1, 5.2
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** AIM
Description:

No definition of north in the orientation of the plant. Recommend:

entity building
…
plant_north
…
end_entity; --- building

ARM_element	AIM_element
plant_north	plane_angle_measure

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Bill/Jay
Accept this as a requirement. Plant north generally doesn't agree with true north. Plant north should be an attribute of the site.

Use gis_position as the basis for defining plant_north?

AI - WB/JR - Review ARM and propose a solution for specifying plant north in relation to gis_position.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 66 **Alt. Numbers:** USA-227-DIS-024 **Issue Date:** 18-Jun-97
Issue Title: Addition of wall sleeve and attributes
Issue Owner: Kenji Araki **Country:** USA
Doc number SC4 N580 **Clause(s):** 4.2
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** ARM
Description:

Wall Sleve with attributes usage, filler_material, shielding_type is suggested.

Proposal **Date:** 24-Sep-97 **Proposer:** Implementers Forum
Rather than create a new ARM entity, the Implementers recommended that the existing entity Reserved_space, which is a subtype of Plant_item, be used to cover the geometry of penetration_hole. Plant_item_collection can be used to define the relationship between penetration_hole and the wall it goes through. Categorization could be on the Plant_item itself. Something should be added to the implementers guide that explains this logic and how the penetration_hole attributes are covered. Add an indication in the AIM element product_definition_relationship.description that the penetration_hole plant_item is a subtraction from the wall it goes through.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mark/Bill/Jay
Describe that one can model the constituents of a penetration if you want to. Provide a worked example to show how this would be done.

AI - MP/WB/JR - Work on example for how penetrations can be handled by the AP 227 model. Check with Kenji Araki for input to example.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 67 **Alt. Numbers:** USA-227-DIS-025 **Issue Date:** 18-Jun-97

Issue Title: Functional vs. Physical Modeling, Relationship with in-line components

Issue Owner: Jean-Christophe/Ian Matthew **Country:** USA

Doc number SC4 N580 **Clause(s):** 4.2

Issue status: open **Classification** Minor Technical

Source: DIS Balloting **Subject:** ARM

Description:

Relationship between line and line_segment. There is an inconsistency between the demo file and other company's interpretations of the breaking up of the lines into line_segments.

Comment **Date:** 18-Jun-97 **Commentor:** Implementers **Action:**

Should we set specific breaking up of lines into line segments for various uses and to handle in-line components? If so, this could be added to the ARM or to the Implementation Guide.

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**

The definition of line and line_segment is arbitrary. The only absolute rules for the uses of lines and line_segments are:

1. A line shall terminate at a piece of equipment.
2. There will be a breakpoint between line_segments where the stream characteristics change. Guidance for the usage of line/line_segment is provided in Annex K, clause K.7.

Comment **Date:** 24-Sep-97 **Commentor:** Implementers **Action:**

Look at AP 221 to see how they handle this issue.

Comment **Date:** 24-Sep-97 **Commentor:** Jean-Christop **Action:**

The second item in the Mitch Gilbert comment indicates that there should be a breakpoint where stream characteristics change. An example of where stream characteristics change is when a reducer is used, but a line change may not be defined at the reducer.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Bill/Mitch

There are no rules for breaking lines into line segments. AP 221 has a set of rules for breaking down lines. We have something in Annex K.

AI - WB - Look at what we can do to clarify the write-up in annex K on rules for breaking line segments or make it more useful for explaining how to handle this.

Issue - JR - Are these things their own product? Using subtypes of product_definitions for various non-plant item items and not having any guidance on what products they are and how do you differentiate them from plant_item items. If yes, then we need to get some recognition criteria put into plant_item to differentiate real plant items from these functional type items. While plant_items are products, not all products are plant_items. This needs to be explicitly stated in the mapping tables, so that we can unambiguously parse a mapping table. We need to be able to map back and forth between the ARM and AIM!!!

AI - MG - Improvement of mapping table - Differentiation of plant_item vs. other product_definition types needs to be made using some kind of mapping rules. What I am thinking is to use the application_context_element for the product_context.

Issue - JR - Mappings are silent as to whose product plant_system belongs to.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 68 **Alt. Numbers:** USA-227-DIS-026 **Issue Date:** 18-Jun-97

Issue Title: Change

Issue Owner: Jean-Christophe/Ian Matthew **Country:** USA

Doc number SC4 N580 **Clause(s):** 4.2

Issue status: open **Classification** Minor Technical

Source: DIS Balloting **Subject:** ARM

Description:

Questions that need to be considered:

How is change used in the current ARM?
How will change_items be carried over in later files, later revisions, and to all collaborators?
Does this satisfy different needs of the industry?

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**

Change management and change history is out of the scope of AP 227. AP 227 is intended for snapshot data exchanges. The change mechanism is to illustrate the changes between states of snapshots.

Comment **Date:** 24-Sep-97 **Commentor:** Implementers **Action:**

What is the practice we should use to so that we don't need to send the whole file when we change something? Do we just send what changed? The practice should be documented in the implementers guide.

Is this issue now addressed by the information Kathy put into the Demo file? Jean-Louis is looking at the revised demo file to determine if the issue is resolved.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Steve/Mitch/Bill/
AI - SK - Use the following discussion from the workshop as the basis for changes to the Scope.

Within scope
· design change information
· change identification/notification - content and predecessor/successor
- original, modified
- date, responsibility
· focus is on comparison of exactly two design snapshots. A single set of changes shall not include multiple changes to a single change_item
· change status across the life cycle of a change
· an item for which a change has been identified shall participate in only one change_item

Out of scope
· management of the complete design history
· change management
· includes policies and procedures and compliance to policies and procedures
· accumulation of change which involves 2 or more successive changes to the same plant item

AI - MG/WB/SK - Investigate the addition of a rule to the AIM such that a change_item that is a "change to" cannot be referenced as a "change from". All changes should be between two design snapshots. Change the one to many relationships between items and changed items to zero to one.

NS - Responsibility of translator or CAD system to merge the changes.

AI - ? - We need to add something to the technical discussion in the AP to cover what happens when we are adding a new part to a system or a new system to the plant, e.g., changed_plant_system, changed_plant.

View of the AP is to show the changes made to what you gave me to start with.

AI - SK - Change structures should be used at the lowest level (highest level of specificity) possible (e.g., use changed_instrument, not changed_plant_item). Cover in 4 and technical discussions.

AI - MG/WB - We need to add a constraint to the ARM that limits the changes to life cycle stages to a single change. The change doesn't become a new change when the life cycle stage changes. Refers to change_pair.

AI - ? - Develop an example of to show how the following types of changes and statuses would be handled and include in the technical discussions annex.

Change Example
· Just proposed change
· Accepted change
· Single component being replaced with multiple components
· Use of change_life_cycle_stage (assuming only comparison of two design snapshots?)

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 69 **Alt. Numbers:** USA-227-DIS-027 **Issue Date:** 18-Jun-97
Issue Title: One to Many Relationships, Reuse of AIM Entities

Issue Owner: Jay Roberts, Manoj Dharwadkar **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1, 5.2
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:
When AIM entities should not be reused, can there be something stated in the implementor's guide or somewhere? Is there a way to expand the reference path, so that it is annotated to say which items either come from another ARM entity (using bold or something) or to actually state which ARM entity that a particular AIM entity in a reference path comes from? This may help in reducing certain duplications and ambiguities.

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**
It is difficult to understand what is being questioned here without a concrete example. Proposals for improvement to the methodology can always be made and, even if not accepted or their implementation is delayed, a usage or implementation guide can be maintained outside of the standardizatation effort which contains the revised mapping tables if that is something PlantSTEP views as desirable and a priority.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Steve/Mar
Need to assess what might be feasible. Distinguish between instances of the same thing and what parts can be reused. There needs to be some additional guidance provided on when to make things and when to use what is already available in the mappings. This is stuff that should be boilerplate in 5.1.

Mapping table clarity issue.

AI - MG - Assess what we can do and put together a write-up for the narrative portion of clause 5 that gives some guidance on how to reuse AIM instances or what can be reused. JR review and critique write-up for coverage of different types of problems.

AI - SK/MP - Take any recommendations to revise the mapping table specification back to QC, mapping table guidelines.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 70 **Alt. Numbers:** USA-227-DIS-028 **Issue Date:** 08-Jul-97
Issue Title: Outside_and_thickness Reference Path
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification** Editorial
Source: DIS Balloting **Subject:** Mapping Table

Description:
The reference path appears to be truncated for Case #2 for the OUTSIDE_AND_THICKNESS Application Element in the mapping tables (page 362).

I believe it should have the additional lines:

characterized object =>
piping_component_class

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The issue is incorrect, it should read:

characterized_object =>
piping_component_class]})

This is another A4 cutoff problem.

Comment **Date:** 17-Oct-97 **Commentor:** Steve Kline **Action:**
Revised AP per the Proposed Solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/23/97 **Impl Resp:** Steve Kline

A4 problem.

AI - SK - Correct the mapping table as described in the proposed solution to the issue.

Issue Number: 71 **Alt. Numbers:** USA-227-DIS-029 **Issue Date:** 09-Jul-97
Issue Title: Schedule / Piping_size_description Mapping
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:

In the AP227 ARM, piping_size_description.dimensional_standard is type plant_spatial_configuration.document.

When we map this to the AIM, we have to create a plant_spatial_document_reference, and we make the assignment:

plant_spatial_document_reference.assigned_document :=
piping_size_description.dimensional_standard.

Later, we map the subtype schedule. It has an attribute schedule.pipe_schedule, which maps to document_usage_constraint.subject_element_value. All document_usage_constraints have an attribute source, which points to a document_reference. In the reference path, it is shown as pointing to this, then subtype plant_spatial_configuration_document reference.

Our assumption is that this is the same plant_spatial_configuration_document_reference whose assigned_document points to the supertype attribute piping_size_description.dimensional_standard for this instance of schedule.

Proposal **Date:** 09-Jul-97 **Proposer:** Kathy Tan

On p.401 of the AP227 DIS, schedule.pipe_schedule has the following reference_path representation:

plant_spatial_configuration_document_item = representation
plant_spatial_configuration_document_item <-
plant_spatial_configuration_document_reference.items[i]
plant_spatial_configuration_document_reference <=
document_reference <-
document_usage_constraint.source
document_usage_constraint
document_usage_constraint.subject_element_value

I believe this reference path is in error, since document_usage_constraint.source should refer to an entity_type document (p.763, and part 41 - p.54), not to the entity_type document_reference.

Beginning from the 6th line, the reference path should probably change to:

document_reference
document_reference.assigned_document ->
document <-
document_usage_constraint.source
document_usage_constraint
document_usage_constraint.subject_element_value

These changes should also be made in schedule's reference path on p. 399.

From what I understand of the use of document_usage_constraint and the use of dimensional standard, I believe that pipe_schedule's document_usage_constraint.source should refer to the dimensional standard document originally created to satisfy piping_size_description's reference_path.

Pipe_schedule's document_usage_constraint.source should refer to the dimensional standard document originally created to satisfy piping_size_description's reference_path. Since the definition of pipe_schedule is by reference to the dimensional_standard, the rule to constrain the value of document type to dimensional standard' needs to be added.

Comment **Date:** 20-Oct-97 **Commentor:** Steve Kline **Action:**
Revised the AP to incorporate the Proposed Solution.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 7/9/97 **Impl Resp:** Steve Kline

Agree. Incorporate the proposed resolution.

AI - SK - Correct the mapping table as described in the proposed solution to the issue.

Issue Number: 72 **Alt. Numbers:** USA-227-DIS-030 **Issue Date:** 11-Jul-97
Issue Title: Outline_shape Reference Path
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:

The reference path for outline_shape seems wrong/incomplete.

First, it probably needs to be of type property_definition. Second, I am making the assumption that representation.name = 'outline' refers to the shape_representation of its is_defined_using relationship.

Proposal **Date:** 24-Sep-97 **Proposer:** Mitch Gilbert

See p. 594. The ARM Plant_item_shape object is the supertype of this object and maps to product_definition_shape. This issue seems to recognize that the subtypes are classifications of the general shape property represented by the plant_item_shape entity. Therefore, mapping of the entities should be to product_definition_shape with a mapping rule that the product_definition_shape is referenced by a group w/ the appropriate .name for each of the different subtypes.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Bill

ARM has been changed.

AI - MG/WB - Clarify for SK how this issue was/is to be resolved.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 73 **Alt. Numbers:** USA-227-DIS-031 **Issue Date:** 12-Jul-97

Issue Title: Need for Sub-typing of Representation

Issue Owner: Jay Roberts **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.2, Annex A

Issue status: open **Classification**

Source: DIS Balloting **Subject:** AIM

Description:

The AIM element for SHAPE_REPRESENTATION_ELEMENT_USAGE lists several options:

ADVANCED_CSG_SHAPE_REPRESENTATION,
PIPING_DESIGN_CSG_SHAPE_REPRESENTATION,
HYBRID_SHAPE_REPRESENTATION,

which are all subtypes of REPRESENTATION. Doing an ARM to AIM mapping, it is unclear which subtype to instantiate. From my reading, just creating an instance of REPRESENTATION should work, and there is no necessity to create this as one of its subtypes.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

There should be more specific text definitions given to the advanced_csg_shape_representation and piping_design_csg_shape_representation subtypes in Clause 5.

The definition for Advanced_csg_shape_representation (5.2.3.1.1, p. 637) says that it "is composed of csg_primitives and boolean operators." Where rule 1 specifies that the representation "shall be a csg_solid, extruded_area_solid, revolved_area_solid, axis2_placement_3d, or mapped_item."

The definition for piping_design_csg_shape_representation (5.2.3.1.42, p. 659) says that it "is composed of CSG primitives and boolean operators and is used to represent the shape of the piping design." Where Rule 1 specifies that the representation "shall consist only of representation_items that are boolean_result, csg_solid, axis2_placement_3d, or mapped_item. "

The two text definitions don't explain the differences, but there are some differences which can be noted in the Express entity definitions.

Also, the text definitions mention CSG primitives, but the where rule 1 of those two representations do not specify the following CSG primitives as possible representation_items: block, right_circular_cylinder, right_circular_cone, sphere, and torus.

There are specific subtypes of representation that must be instantiated for the mapping; an instantiation of the supertype representation does not satisfy the requirements as specified in the ARM. Which subtype that should be instantiated cannot be specified in the mapping table because it is dependent on the data instances that are being represented in the physical file. Therefore, the issue becomes one against the definitions in clause 5.2 as Kathy has pointed out. The fact that there are other solids in the shape that are not CSG is not reflected in the definition. What if there are no CSG solids in the representation, then the definition is completely misleading. What is also not clear is that if it contains CSG primitives and boolean operators by means of the csg_solid. The definitions of the advanced_csg_shape_representation and piping_design_csg_shape_representation entities need to be reviewed and clarified.

WR1 doesn't mention the csg primitives due to the fact that they are included through the csg_solid entity. The real difference between the advanced_csg_shape-reprsentation and the piping_design_csg_shape_representation is that the extrusions are not included in the piping_design_csg_shape_representation, and the piping_design is a little looser since it can simply have a boolean_result as an item without formally having a csg_model. These differences should be highlighted in the piping_design_csg_shape_representation entity definition.

In any event, the only valid instantiation for representation is for the parametric representation which is also questionable given the fact that it is a parametric shape representation. The question that must be answered to determine if a mapping change is warranted is, does the parameteric representation with its current scope have a coordinate space dimension, or is it simply a listing of parameters? If so, the OR case here for parametric representation (the representation containing shape_parameters) mapping should be to shape_representation as well with the same mapping rule on the representation.name.

Comment **Date:** 25-Sep-97 **Commentor:** Implementers **Action:**

Improve the definitions in 5.2.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Bill/Mitch/Steve

Only plant_csg and hybrid remain as types of shapes in ARM. Covered by other discussions.

AI - WB/MG/SK - ARM changes - Add plant_csg and hybrid_shape_representation under shape_representation. This should be a complete categorization, but we need to check for any other ramifications. Move outline, detailed, and envelope shape to a different category tree under shape_representation.

Issue - Do the changes described above affect the resolution of Issue No. 72 (US-227-DIS-030)?

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 74 **Alt. Numbers:** USA-227-DIS-032 **Issue Date:** 13-Jul-97

Issue Title: Line_to_line_connection to Line_to_line_termination Mapping

Issue Owner: Jay Roberts **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: open **Classification**

Source: DIS Balloting **Subject:** Mapping Table

Description:

In the mapping for the relationship LINE_TO_LINE_CONNECTION to LINE_TO_LINE_TERMINATION,

shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(connection_node)
(plant_line_segment_termination)

And

shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(plant_line_segment_termination)

Both paths are to a LINE_TO_LINE_TERMINATION. What is the significance of the (connection_node) in the first path?

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The connection_node is created when there are more than two lines being terminated at a single point. There is a 2-N cardinality for line_to_line_connection to line_to_line_termination. The most common case is the 2 cardinality, but there are branches where more than 2 lines are terminated. For the 2 case, the line_to_line_connection is simply a connection relationship between 2 line_to_line_terminations. For the more than two, there needs to be a shape_aspect that models the connection_node (that is the logical connection point) shared by all of the terminations. That is the reason for the OR.

This discussion needs to be in the definition of connection_point in some form.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch

Add to usage guide? No, add to 5.2 under connection_node.

AI - MG - Include the discussion in the proposed solution in the definition of the connection_point entity in the AIM.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 75 **Alt. Numbers:** USA-227-DIS-033 **Issue Date:** 16-Jul-97

Issue Title: Site / Sited_plant / Planned_physical_plant Relationships

Issue Owner: Jay Roberts **Country:** USA

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: resolved **Classification**

Source: DIS Balloting **Subject:** Mapping Table

Description:

My interpretation of the mapping table implies that only one PLANNED_PHYSICAL_PLANT can be associated with one SITE but many SITEs can be associated with the same PLANNED_PHYSICAL_PLANT. This is the exact reverse of what I would expect, but I have followed the following train of logic:

This is because:

AIM element for SITE:

site<=
[characterized object]
[property_definition]

AIM element for SITED_PLANT:

property_definition.definition

AIM element for PLANNED_PHYSICAL_PLANT:

product_definition

Relationships:

SITE to SITED_PLANT:

site <=
property_definition
property_definition.definition

PLANNED_PHYSICAL_PLANT to SITED_PLANT:

product_definition
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition

Hence:

SITE to PLANNED_PHYSICAL_PLANT

site <=
property_definition
property_definition.definition->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition
characterized_product_definition = product_definition
product_definition

Which is a M:1 relationship. Any insight on this would be appreciated.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The mapping for sited_plant should be modified to a subtype of property_definition as follows:

```
ENTITY sited_plant
  SUBTYPE OF (property_definition);
UNIQUE
  UR1: SELF.definition;
WHERE
  WR1: 'PLANT_SPATIAL_CONFIG_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
       (SELF.definition);
  WR2: SELF.definition.frame_of_reference.name = 'physical_occurrance';
```

END_ENTITY;

The UNIQUE rule will enforce the Z cardinality from planned_physical_plant to sited_plant and the WHERE rules will enforce the exactly one cardinality from sited_plant to planned_physical_plant.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert
AI - MG - Review the mapping of Site in light of the requirement of it's M:1 relationship to Plant.

Issue Number: 76 **Alt. Numbers:** USA-227-DIS-034 **Issue Date:** 18-Jul-97
Issue Title: Subtypes of Plant_item_shape
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
AIM elements for DETAIL_SHAPE,ENVELOPE_SHAPE, and OUTLINE_SHAPE are

representation.name = 'detail'
representation.name = 'envelope'
representation.name = 'outline'

The AIM element for the base class, PLANT_ITEM_SHAPE is:

PRODUCT_DEFINITION_SHAPE

The only AIM REPRESENTATION entity that seems relevant in the mapping tables is the AIM element for SHAPE_REPRESENTATION_ELEMENT_USAGE. However, according to the information requirements for this entity, it is already assigned a different .name attribute.

The question is, then, to which representation does the mapping tables for the subtypes of PLANT_ITEM_SHAPE refer?

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum
Refer to Issue No. 72 (USA-227-DIS-030).

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch
Same as US - 30.

JR - Cardinalities between shape_representation, shape_representation_element_usage, shape_representation_element, plant_item_shape, and possibly reference_geometry and interfering_shape_element need to be looked at.

AI - MG - Shape_representation_element_usage needs to be mapped to gri. The relationship between shape_representation_element and the usage will be identical mapping.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 77 **Alt. Numbers:** USA-227-DIS-035 **Issue Date:** 21-Jul-97
Issue Title: Use of Calendar_date versus Date
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: resolved **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
Tthe AIM element that the following attributes are mapped to should be changed:

change.date (ARM 4.2.15.5, mapping table p. 164)
change_item.creation_date (ARM 4.2.18.2, mapping p. 169)
change_life_cycle_stage_usage.date_of_activation (ARM 4.2.21.1, mapping p. 177)
change_life_cycle_stage_usage.date_of_completion (ARM 4.2.21.2, mapping p.178)

ARM definition 4.2.15.5 defines the change.date as "the calendar day-month-year and time that the Change was initiated on," but the AIM element date only has a field for year. Only the subtype calendar_date will give the day-month-year, as specified in the ARM definition.

To obey the ARM definitions, they should all map to the AIM element calendar_date (part 41 - 4.9.4.2) or date_and_time (part 41 - 4.9.4.7) where the attribute date_component is a calendar_date. If they are mapped to calendar_date, the requirement for time should be removed from the ARM definition. If they are mapped to date_and_time, the attribute change_approval.approval_date should also be mapped to date_and_time; currently, it is mapped to calendar_date.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert
The mapping of the attributes listed in the issue should be changed to calendar_date OR date_and_time. According to the AE statement, the time is not always known, hence the mapping to calendar_date. This will require a change to the AIM and EXPRESS-G for the date_and_time assignment select and to several rules for required dates to require a date or a date_and_time.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert
AI - MG - Incorporate the proposed solution.

Issue Number: 78 **Alt. Numbers:** USA-227-DIS-036 **Issue Date:** 25-Jul-97
Issue Title: Incomplete Reference Path for Change.date
Issue Owner: Kathy Tan **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: resolved **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
On page 164 of the Mapping Table (Table 2 - change information UoF), the change.date reference path is missing a reference to date_role; date_assignment.role is a required attribute which takes an entity type date_role. change_item.creation_date, change_life_cycle_stage_usage.date_of_activation, and change_life_cycle_stage_usage.date_of_completion all include date_role in their reference path.

The reference should look like:

change_action

plant_spatial_configuration_dated_item <-
plant_spatial_configuration_date_assignment.items [i]
plant_spatial_configuration_date_assignment <=
date_assignment
{date_assignment.role ->
date_role
date_role.name = '<some specified name such as initiation date or this line
can be left out>'}
date_assignment.assigned_date ->
date

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The only reason to include a rule for the date_role is to specify what value must always be given to the name attribute. Therefore, if a predefined value is desired, a mapping rule can be added.

Add something to the implementers guide that indicates that this field should be "change_date" since that is the only thing we could come up with now, but other values may be identified at a later point. The role of the association of date to the change.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**

Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

AI - MG - Incorporate the proposed solution.

Issue Number: 79 **Alt. Numbers:** USA-227-DIS-037 **Issue Date:** 18-Aug-97
Issue Title: Shape_representation versus Representation to Represent a Shape
Issue Owner: Nikolay Shulga **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:

Apparently, it is possible to represent a shape in AP227 by using representation entity rather than shape_representation entity, as long as name attribute is set to one of the several strings, e.g.,

#666 = representation ('parametric shape representation', (#20, #30));

means that it is really a shape.

I have a few problems with that.

1) This is very unefficient way of storing information
2) This is very unefficient way to retrieve information: string comparison is costly on most non-Intel platforms.
3) This is misleading as we also have shape_representation arguably to do the same thing: represent shapes.

There should be only one way to represent shapes. Either always use shape_representation (I'd prever that, as this is clearer and consistent with other APs), or get rid of shape_rep and always use representation with specified name.

Quite apart from making a change in the standard, we need to agree on what we do in the implementations, I guess we can do it through an implementors' agreement.

Comment **Date:** 18-Aug-97 **Commentor:** Bill Burkett (**Action:**

[response to Nikolay Shulga email]

It is my own personal opinion that EXPRESS-G is nearly as bad a choice for ARMs as IDEF1X. My reason for this opinion is difficult to explain, but it has to do with the way people use data modelling languages and the knowledge and experience they draw upon when doing so. If EXPRESS-G ARMs were developed by Users/Domain Experts (who *should* be developing the ARM and who do not design databases for a living), the implementors of an AP will more-than-likely think it's a terrible model (which it probably will be) and wonder why they don't just use entities from the Integrated Resources to create the ARM - it would make writing translators so much easier. The implementors will lose sight of the need for the ARM. (In fact, I think it is this perspective that lead to the whole "implementable ARM" debate.)

If, instead, AP implementors develop an EXPRESS-G ARM, it will be designed from their perspective of the data they have in their system and want to exchange. While treating an existing data store as used to support a CAD system, for example, as a statement of requirements is a perfectly fine and reasonable position, it does not really account for the requirements of the *users* CAD systems within the domain addressed by the system.

The role of the ARM and the AIM are very different. I feel that using the same modelling language muffles this distinction. My own personal recommendation would be to use a vanilla ER modelling language for ARMs, such as that described in the Elmasri/Navathe database textbook.

Comment **Date:** 18-Aug-97 **Commentor:** Nikolay Shul **Action:**

[response to Manoj Dharwadkar email]

The solution proposed for inclusion in the User's Guide works for me.

Comment **Date:** 18-Aug-97 **Commentor:** Manoj Dharw **Action:**

In my opinion, mapping of ARM Entity SHAPE_REPRESENTATION_ELEMENT_USAGE to AIM Entity Representation (where representation.name = 'parametric shape representation') has been introduced only to support the ARM Entity SHAPE_PARAMETER (subtype of SHAPE_REPRESENTATION_ELEMENT). The SHAPE_PARAMETER entity is introduced in the DIS version.

I agree with Nikolay that there should be consistency in mapping the ARM Entity SHAPE_REPRESENTATION_ELEMENT_USAGE and it should map to AIM Entity SHAPE_REPRESENTATION.

In the current DIS mapping table (page 603), the allowable AIM Entity for ARM Entity SHAPE_REPRESENTATION_ELEMENT_USAGE are:

1) ADVANCED_CSG_SHAPE_REPRESENTATION
2) PIPING_DESIGN_CSG_SHAPE_REPRESENTATION
3) HYBRID_SHAPE_REPRESENTATION
4) REPRESENTATION, where REPRESENTATION.NAME='parametric shape representation'

Cases 1,2,3 are all subtypes of SHAPE_REPRESENTATION.

Why is SHAPE_REPRESENTATION not listed as a valid AIM ENTITY?

The PlantSTEP demo file uses SHAPE_REPRESENTATION to represent all the geometry. We have not seen the use of 1,2,3 or 4 in the PlantSTEP demo file.

I propose the following solution to be included in the Implementor's User guide:

The allowable AIM Entities for ARM Entity SHAPE_REPRESENTATION_ELEMENT_USAGE are:

1) ADVANCED_CSG_SHAPE_REPRESENTATION
2) PIPING_DESIGN_CSG_SHAPE_REPRESENTATION
3) HYBRID_SHAPE_REPRESENTATION
4) SHAPE_REPRESENTATION

The constraint in the reference path (representation.name = 'parametric shape representation') should be discarded. The reference path for ARM Entity SHAPE_PARAMETER specifies this constraint.

Comment **Date:** 18-Aug-97 **Commentor:** Nikolay Shul **Action:**

[response to Mitch Gilbert email]

Yes, it would be helpful. But the long-term solution is computer-readable mapping table.

Comment **Date:** 18-Aug-97 **Commentor:** Mitch Gilbert **Action:**

[response to a portion of Bob Fisher email]

> I think my use of the word "code" was misleading: I simply meant the EXPRESS in the ASCII data file, which is the end result of the rules in the mapping table. We are talking about the same thing. I think of the ASCII files as code, because they are more complex than just a set of formatted data fields.
>

As far as I recall, EXPRESS-M doesn't meet the requirements to do the job. There is no capabililty to map to a constrained reference path among other things. EXPRESS-X has been taking on some of these requirements, but its development is still in its infancy. BTW, I agree with you folks, in principle, that the mapping table ought to be computer sensible. There has been work undertaken to write the BNF for the reference path syntax, but the funding was cut. It is a matter of practicality at this point.

Mark Palmer and I have been discussing using an editorial convention (bold face) for the important areas (where there are value constraints, for example) in the table. Do you think this kind of convention would help in your work?

Comment **Date:** 18-Aug-97 **Commentor:** Nikolay Shul **Action:**

I think eventually ARM Express should be come mandatory, EXPRESS-X mapping table as well. IDEF1X/Express-G/etc generated from the above. Ideally, US TAG should be pushing for this now.

Well, back to shape_rep - any opinions as to what we should use? Does anybody have any recollection as to how representation("whatever name"()) representing a shape got there in the first place? AP221 or just PDIT?

Comment **Date:** 18-Aug-97 **Commentor:** Kent Reed (e **Action:**

[response to portion of Bob Fisher email]

>I think we all are "very uneasy" about the useability of paper mapping tables. Variants of EXPRESS-M (EXPRESS-X conformance class one) are commercially available today: with an implementor's agreement, could they be used? I think so... Main problem is that most implementors see it as necessary to work "through" an ARM level schema (whether the mapping itself is done in one or two passes), but the need for ARM EXPRESS is not acknowledged by the standard, so we would have to come up with (and agree amongst ourselves) on that as a pre-requisite to any EXPRESS-X to replace the mapping tables.

It's not that "the standard" doesn't acknowledge ARMs in EXPRESS, it's that the AP227 team used IDEF1X rather than EXPRESS-G for the graphical ARM notation, as did other teams. There were good and sufficient reasons for that decision.

Nevertheless, I have been thinking for some time that having an agreed EXPRESS translation of the IDEF1X ARM would be a "good thing" for the very reasons that Bob and Nikolai are raising, and I would be willing to contribute to the creation of such a thing.

Of course, as Mark pointed out to me just now, having more than one representation of the ARM makes configuration management a nightmare, so perhaps we should define the EXPRESS version as the baseline for the second edition AP in anticipation of retiring the IDEF1X version (just a thought!).

Comment **Date:** 18-Aug-97 **Commentor:** Kent Reed (e **Action:**
I can't decide which of Nikolai's points I agree with more strongly: 1) having only one way to indicate shape representation or 2) wanting a computer-sensible mapping table notation. Mind you, this is my personal opinion, not a legal one!

As for 1), an implementors' agreement does seem called for, once we understand what the mapping table is trying to tell us and why.

As for 2), it appears that EXPRESS-M has fallen short, and EXPRESS-X is still in gestation. In the meantime, I know a contractor was writing some code for the Application Protocol Development Environment project elsewhere in NIST that was supposed to parse the existing mapping-table notation for testing and verification purposes. I'll have to ask what has happened with that project.

Comment **Date:** 18-Aug-97 **Commentor:** Bob Fisher (e **Action:**
I think my use of the word "code" was misleading: I simply meant the EXPRESS in the ASCII data file, which is the end result of the rules in the mapping table. We are talking about the same thing. I think of the ASCII files as code, because they are more complex than just a set of formatted data fields.

I think we all are "very uneasy" about the useability of paper mapping tables. Variants of EXPRESS-M (EXPRESS-X conformance class one) are commercially available today: with an implementor's agreement, could they be used? I think so... Main problem is that most implementors see it as necessary to work "through" an ARM level schema (whether the mapping itself is done in one or two passes), but the need for ARM EXPRESS is not acknowledged by the standard, so we would have to come up with (and agree amongst ourselves) on that as a pre-requisite to any EXPRESS-X to replace the mapping tables.

Comment **Date:** 18-Aug-97 **Commentor:** Nikolay Shul **Action:**
The problem is, this isn't Express code - this is mapping table. Which allows for anything.

Incidentally, I am bcoming increasingly uncomfortable with the idea of non-computer-readable mapping table..

Comment **Date:** 18-Aug-97 **Commentor:** Bob Fisher (e **Action:**
My opinion is that it is mandatory to either:

a) have only one way to represent in the EXPRESS code any specific information aspect or b) if it "must be" possible to model a specific information aspect in different ways (because each of those ways has to exist for similar but different aspects), then the standard needs to be explicit as to how each aspect is to be modeled. I don't believe the current format allows for this option.

Comment **Date:** 19-Aug-97 **Commentor:** Nikolay Shul **Action:**
[response to Bill Burkett email]

I think you missed the point. Nobody is asking to get rid of the text in the ARM - IMO, this is the most valuable part of the standard. But it should be augmented by an Express (not Express-G; does your opinion apply to Express as well?) schema. And then the mapping table format should be changed so it is as computer-readable as well.

I think the structure of labour costs in development of STEP s/w is such that this will happen anyway - on s/w vendor level, if nowhere else. Which will be unfortunate as it leads to 1) same work done at many different places, at a cost to end users of our s/w 2) discrepancies in interpretation.

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**
The mapping should be to shape_representation with the same constraint on the name for the parametric shape. This issue needs to be raised against the DIS document.

Comment **Date:** 24-Sep-97 **Commentor:** Implementers **Action:** Mitch Gilbert
Can a Hybrid_shape_representation have a shape_parameter in it? Yes, per Mitch Gibert. Use Hybrid_shape_representation instead of shape_representation. How do you know which of the options to use? You need to go to the shape_representation_element_usage to see what option is required based on the shape_representation_element.

AI - Why was shape_representation excluded from the mapping? Mitch Gilbert to check with Diane Craig to see why.

There is not an explicit requirement in the ARM for a CSG model. It is implicit from the csg_element entity in the ARM.

P. 248 - Eccentric_cone and eccentric_cylinder have the same mapping. Why? Representation_item.name = eccentric.cone should be added to mapping. If we change to the proposed solution to Issue No. 601, then nothing needs to be done, because this will take care of the concern. Jacobus will use the representation_item.name solution for the demo in the interim.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch Gilbert
AI - MG - Check if the shape_parameter would have a 2d or 3d context. If so, it should be mapped to shape_representation.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 80 **Alt. Numbers:** USA-227-DIS-038 **Issue Date:** 09-Sep-97
Issue Title: Identification of AIM Instance Values of "in situ" and "placed" in ARM Object Plant_item_loca
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 4.2, 5.1, Annex G
Issue status: resolved **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
It is unclear from the mapping tables how the ARM differentiates between "in-situ shape" and "placed shape" for plant_item_location.

There is a descriptive difference, and it is clear how to differentiate between AIM instances. However, given an ARM instance of plant_item_location, it is unclear how we would determine whether it was "in situ" or "placed".

If this cannot be determined from the ARM alone, our opinion is that this is a deficiency that needs to be resolved.

Proposal **Date:** 24-Sep-97 **Proposer:** Implementers Forum
If something doesn't have an origin, then its an in-situ shape. Its geometric_representation_context is the same as the Plant or Building, etc. that is in-situ in. Things that are placed should have their own geometric_representation_context. Placing something calls for a transformation between coordinate systems using location and orientation. If you have a plant_item_definition then you have a placed shape.

Need to document the discussion in the implementers guide. This is a fundamental concept.

Comment **Date:** 24-Sep-97 **Commentor:** Implementers **Action:** SWK/Implemente
AI - Need to look at the NIST demo file. We apparently use the in-situ and our items should be placed and refer to their own coordinate system and not the plant coordinate system. Need to change the demo file by adding a context for each shape_representation with its own coordinate system.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Steve Kline
AI - SK - Include Mitch's write-up of in-situ vs placed shape in 5.2.1.

Issue Number: 81 **Alt. Numbers:** USA-227-DIS-039 **Issue Date:** 09-Sep-97
Issue Title: Value of AIM Element Document.kind
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
The mapping table shows the aim element for STRUCTURAL_COMPONENT.SIZE_DESIGNATOR is DOCUMENT. However, no value or guidance is provided for what DOCUMENT.KIND should be. Our assumption is that this AIM element must be created on an ARM to AIM translation and is not a pre-existing element.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert
If there is a standard type of document here then it should be stated in a rule. The type of document depends on many factors. It depends on the type of material, industry of application, etc.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch Gilbert
AI - MG - Investigate if there is a standard type of document for the structural_component.size_designator.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 82 **Alt. Numbers:** USA-227-DIS-040 **Issue Date:** 09-Sep-97
Issue Title: Frame_of_reference for Product_definition
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
The AIM element for REQUIRED_MATERIAL_DESCRIPTION is PRODUCT_DEFINITION. What is the FRAME_OF_REFERENCE for these product definitions?

Comment **Date:** 23-Sep-97 **Commentor:** Mitch Gilbert **Action:**
The required_material_description could have any context depending on the life cycle stage at which the requirement is defined. It could be defined in the manufacturing planning stage, or it could be a critical item for which the material requirement is identified during conceptual process design. I don't know if we need to standardize it.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch Gilbert
AI - MG - Define mapping rules for product_definition_context for these elements

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 83 **Alt. Numbers:** USA-227-DIS-041 **Issue Date:** 15-Sep-97
Issue Title: Plant_item_connector Ambiguities
Issue Owner: Jay Roberts **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: resolved **Classification**
Source: DIS Balloting **Subject:** Mapping Table
Description:
1) There seems to be ambiguity in the relationship between plant_item_connector_occurrence and physical occurrences/definitions of plant_items.

Discussion:

The AIM element for an ARM plant_item_connector also named plant_item_connector, which is a subtype of shape_aspect. The ARM relationship between plant_item and (ARM) plant_item_connector is represented in the AIM by:

(AIM) plant_item_connector.of_shape -> product_definition_shape
product_definition_shape-> product_definition

where the product_definition is the AIM element for the planned_physical_plant_item. A product_definition_shape that points to the AIM element of a physical occurrence/definition ofa plant_item is represented in the ARM as a plant_item_shape.

However, sec 4.3.87 of the DIS documents notes that a plant_item can be described by more than one plant_item_shape.

In the case of multiple plant_item_shapes related to the same physical occurrence/definition, it appears to be indeterminate to which product_definition_shape (AIM element of plant_item_shape) the plant_item_connector should point.

2) Inability to form all relationships for (ARM) shape_representation.

Per sec. 4.3.99, an (ARM) plant_item_connector may be described by zero, one, or many (ARM) shape_representations. The AIM element for an ARM shape_representation is property_definition_representation. The relationship is expressed in the AIM by:

property_definition_representation.definition->plant_item_connector.

Per sec. 4.3.112, each (ARM) shape_representation defines exactly one plant_item_shape. For a (ARM) shape_representation related to a (ARM) plant_item_connector, our reading is that this is also the plant_item_shape whose (AIM) product_definition_shape is pointed to by (AIM) plant_item_connector. In certain cases, this would clear up the ambiguity of section 1) above, except that the relationship (ARM)

shape_representation to plant_item_shape is represented in the AIM as:

property_definition_representation.definition->product_definition_shape.

From this, we see that due to AIM entity limitations, a (ARM) shape_representation can not have a relationship simultaneously with a (ARM) plant_item_connector and with a plant_item_shape. This is in contradiction of section 4.3.112.

Also note that resolving issue 2) does not automatically resolve 1), since the following cases would be allowable for a plant_item with multiple plant_item_shapes even were 2) not an issue:

A) No (ARM) shape_representations associated with a (ARM) plant_item_connector
B) Multiple (ARM) shape_representations pointing to multiple plant_item_shapes associated with the same plant_item.

Proposal **Date:** 23-Sep-97 **Proposer:** Mitch Gilbert

The problem is, generally, that plant_item_shape, shape_representation and shape_representation_element_usage combine concepts of shape and the representation of shape differently than the STEP IRs. This problem makes for a complex mapping where all aspects of each may not be evident, especially in the relationships among the items, their properties and how the properties are represented.

1) the requirement for multiple shapes for a plant item described in the assertion combines (from the viewpoint of the application expert) the STEP concepts of property_definition and representation. That is, the semantics is not that a plant_item in and of itself may have more than one shape, but its shape (the property in STEP terms) may have more than one representation (in STEP terms). This cardinality is maintained, although it is not obvious in the mapping table due to the conceptual separation of the concepts and their positioning within the different entities in both layers.

2) This appears to be an ARM problem. Since plant_item_connector is not a plant_item, the cardinality to plant_item_shape from shape_rep should be Z.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Fixed by changes in the shape stuff in the ARM.

AI - MG - Verify that the cardinalities are correct.

Issue Number: 84 **Alt. Numbers:** USA-227-DIS-042 **Issue Date:** 23-Sep-97
Issue Title: Plant_item_connector to Shape_representation Reference Path
Issue Owner: Mitch Gilbert **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: closed **Classification**
Source: Implementer's Forum **Subject:** Mapping Table

Description:
The reference path for plant_item_connector to shape_representation incorrectly goes past property_definition_representation. Are advanced_csg_shape_representation, piping_design_csg_shape_representation, and hybrid_shape_representation the only types that apply to a plant_item_connector? There are no rules to this effect in the AIM. If they are, these should be placed in a rule. If they are not, the last six lines should be removed.

After further discussion, the latter case is the one that is applicable, the last six lines should be removed from the mapping table (see p. 237). The path should end at property_definition_representation.

Proposal **Date:** 25-Sep-97 **Proposer:** Implementers Forum
Delete the last six lines of the reference path.

Comment **Date:** 17-Oct-97 **Commentor:** Steve Kline **Action:**
Incorporated change into AP.

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to closed to reflect incorporation of proposed solution.

Resolution: **Res Date:** 9/25/97 **Impl Resp:** Steve Kline

Agree with proposed resolution.

AI - SK - Delete the last six lines of the reference path for plant_item_connector to shape_representation as indicated in the proposed solution to the issue.

Issue Number: 85 **Alt. Numbers:** USA-227-DIS-043 **Issue Date:**
Issue Title: Mapping of Plant_item_collection and Changed_plant_item_collection
Issue Owner: **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1 (Table 9)
Issue status: resolved **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:
The Plant_item_collection object is mapped to assembly_component_usage. However, according to the definition in clause 4.2.173, there can be Plant_item_collection objects that are not assembly relationships. An example is a kit. The Connected_collection subtype of Plant_item_collection is specifically defined as an assembly component relationship.

Proposal **Date:** **Proposer:**
Remap the Plant_item_collection and Changed_plant_item_collection objects to product_definition_relationship

Comment **Date:** 04-Jun-98 **Commentor:** Steve Kline **Action:**
Changed from open to resolved per 1998-03-16/18 workshop discussion noted in resolution.

Resolution: **Res Date:** 3/16/98 **Impl Resp:** Mitch Gilbert

Plant_item_collection is not necessarily an assembly. Incorporate the proposed resolution.

AI - MG - Incorporate the proposed solution to the issue.

Issue Number: 86 **Alt. Numbers:** USA-227-DIS-044 **Issue Date:**
Issue Title: Shape_representation_element_usage Mapping Clarification

Issue Owner: **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1 (Table 10), 5.2.3.1.27, 5.2.4.16
Issue status: open **Classification** Editorial
Source: DIS Balloting **Subject:**

Description:

The usage of the shape_representation supertype for the mapping of the shape_representation_element_usage object is unclear. The rule subtype_exclusive_shape_representation allows for the instantiation of the shape_representation supertype without one of its subtypes. This, however, is not for the mapping of the Shape_representation_element_usage object; it is for other areas of the model. The hybrid_shape_representation subtype should be used when the shape is not structured.

Proposal **Date:** **Proposer:**

Clarify the mapping of the Shape_representation_element_usage object with a note.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch Gilbert

Mapping will be made more precise by the new ARM structure.

AI - MG - Revise mapping as needed to reflect the changes covered by other issues to the shape area ARM structure.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 87 **Alt. Numbers:** USA-227-DIS-045 **Issue Date:**
Issue Title: Line_start_location and Line_end_location Attributes
Issue Owner: **Country:** USA
Doc number SC4 N580 **Clause(s):** 4.2.167.2, 4.2.167.3
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:**

Description:

The line_start_location and line_end_location attributes contain two separate pieces of information (the physical location of the line termination and a BOP/COP/TOP indicator) and are misplaced. Each piping system line may have specified exactly one start location and/or exactly one end location. Therefore, the start/end location is actually an attribute of the line as a whole; the reason the attribute is currently part of the line termination is that the start/end location will *correspond* to a termination point.

Proposal **Date:** **Proposer:**

The two separate pieces of information should be separated into two separate attributes and the attributes should be associated with a piping_system_line and with a termination.

Resolution: **Res Date:** 3/16/98 **Impl Resp:**

See Issue No. 54 (US-227-DIS-012).

Issue Number: 88 **Alt. Numbers:** USA-227-DIS-046 **Issue Date:** 14-Nov-97
Issue Title: Mapping Table Syntax Inconsistencies
Issue Owner: Steve Kline **Country:** USA
Doc number SC4 N580 **Clause(s):** 5.1
Issue status: open **Classification** Minor Technical
Source: DIS Balloting **Subject:** Mapping Table

Description:

Review of the mapping table as part of the development of an abstract test suite (ATS) for AP 227 is finding a number of syntactical errors (e.g., unmatched parentheses and braces) and minor inconsistencies in the mapping of similar objects. An example of an inconsistency found is provided below:

The reference path of BRANCH_HOLE.diameter (Table 4 - connector UoF) states in part:

#1: (shape_aspect <-
dimensional_size.applies_to
dimensional_size
dimensional_characteristic = dimensional_size
dimensional_characteristic <-
dimensional_characteristic_representation.dimension
dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation <
shape_representation <
representation
{representation.name = 'piping connector dimensional shape'}
representation.items[i] ->
{representation_item
representation_item.name = 'diameter'})

The reference path for BLANK.outside_diameter (Table 6 - piping component UoF) states in part:

#1: (piping_component_definition <
product_definition
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
shape_aspect <-
dimensional_size.applies_to
dimensional_size
dimensional_characteristic = dimensional_size
dimensional_characteristic <-
dimensional_characteristic_representation.dimension
dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation <
shape_representation <
{representation
representation.name = 'blank fitting dimensional shape'}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'outside diameter'})

The identified inconsistency between these reference paths is the placement of the AIM object "representation" and the placement of the "{" for the constraint involving "representation.name". While

the defferent mappings appear to provide the same meaning, a more rigorous consistency in mappings will help to simplify machine parsing of the reference paths during implementation of the AP.

Comment **Date:** 16-Mar-98 **Commentor:** Workshop **Action:** Mitch/Steve

Work is underway to make the mapping table syntax consistent for rules. Putting the rule at the entrance or exit.

AI - MG/SK - Need to define a consistent way to approach the inclusion of mapping rules - entrance/exit to/from rule. Create a consistent format. Put out a discussion of what the ITI tools require and MG will review for impact.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 89 **Alt. Numbers:** IF-54 **Issue Date:** 09-Dec-97

Issue Title: Functional_plant/Planned_physical_plant vs Functional_design_view/Physical_design_view

Issue Owner: Jay Roberts **Country:**

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: open **Classification**

Source: Implementer's Forum **Subject:** Mapping Table

Description:

The AIM elements for functional_plant and planned_physical_plant are product_definition with frame_of_reference pointing to an appropriate product_definition_context and formation_of_product pointing to a plant entity.

The product_definition/frame_of_reference combination for functional_plant will cause and instantiation of function_design_view and likewise planned_physical_plant will cause instantiation of physical_design_view.

These entities are unrelated to plant, however, so the mapping tables should note that when formation.of_product is plant, physical/function_design_view should not be instantiated.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 90 **Alt. Numbers:** IF-55 **Issue Date:** 11-Dec-97

Issue Title: Planned_physical_plant_item.type Mapping

Issue Owner: Jay Roberts **Country:**

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: open **Classification**

Source: Implementer's Forum **Subject:** Mapping Table

Description:

planned_physical_plant_item::type has the identical mapping path as structural_component::type and as equipment::type.

Therefore, for complex entities such as planned_physical_plant_item_and_equipment, this can only have one value, as there is no way to distinguish which ARM entity the mapping path applies to. Currently, there is no such restriction.

Recommend that all implementations adhere to this practice until this issue can be resolved.

Proposal **Date:** 12-Dec-97 **Proposer:** Jay Roberts

Add the following lines to the respective reference paths:

1) planned_physical_plant_item::type
Add:
"product_category.description = 'planned physical plant item type"'

2) equipment::type
Add:
"product_category.description = 'equipment type"'

3) structural::type
Add:
"product_category.description = 'structural component type"'

Until this issue is resolved within the standard, our AIM files will adhere to this practice.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 91 **Alt. Numbers:** IF-56 **Issue Date:** 19-Nov-97

Issue Title: ARM Shape_representation_element_usage to Shape_representation Inconsistency

Issue Owner: Jay Roberts **Country:**

Doc number SC4 N580 **Clause(s):** 4.2, 5.1

Issue status: open **Classification**

Source: Implementer's Forum **Subject:** Mapping Table

Description:

According to the DIS version of AP227, multiple instances of shape_representation_element_usage may be associated with one shape_representation in the ARM.

However, it is impossible to represent this situation in the AIM due to the nature of the AIM elements for the ARM entities.

The AIM element for shape_representation is PROPERTY_DEFINITION_REPRESENTATION. The AIM element for shape_representation_element_usage is REPRESENTATION or one of it's various subtypes.

The relationship between the two in the ARM is manifested in the AIM as:

PROPERTY_DEFINITION_REPRESENTATION.DEFINITION = REPRESENTATION.

Hence, a given instance of shape_representation in the ARM can only be related to a single instance of shape_representation_element_usage.

Proposal **Date:** 19-Nov-97 **Proposer:** Jay Roberts

Short term: Change the wording of the arm to reflect that this is a singular relationship.

Long term: The ARM representation of geometry is rather intricate, with plant_item_shape, shape_representation, shape_representation_element_usage, shape_representation_element, and reference_geometry. In light of this latest requirement, it becomes very inefficient. Perhaps some rethinking of the geometry elements in the ARM is in order.

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 92 **Alt. Numbers:** IF-57 **Issue Date:** 16-Dec-97

Issue Title: Product for piping_system, plant_line_definition, plant_line_segment_definition

Issue Owner: Jay Roberts **Country:**

Doc number SC4 N580 **Clause(s):** 5.1

Issue status: open **Classification**

Source: Implementer's Forum **Subject:** Mapping Table

Description:

piping_system, plant_line_definition, and plant_line_segment_definition are all subtypes of product_definition. This means that they all have attribute product_definition::formation (product_definition_formation) which then has attribute product_definition_formation::product. This defines the product for which the piping_system, plant_line_definition, and plant_line_segment_definition provide definitions.

In the NIST file, the product that piping_system points to is a unique product entity, and is clearly a "piping_system" product that demonstrates that a piping_system is a product unto itself. However, plant_line_definition and plant_line_segment_definition ultimately point to the plant entity, which is a subtype of product. That is to say, plant_line and plant_line_segment are not considered discrete products whereas piping_system is.

This is somewhat inconsistent, however, if a plant_line is not a discrete product, I would think that it would be a definition of the piping_system product to which it is related rather than the plant product. It is related to the piping_system via a product_definition_relationship, however.

This seems like a murky area. Are there any comments out there on this?

Comment **Date:** 16-Dec-97 **Commentor:** Jay Roberts **Action:**

I was really addressing a less broad specification issue, which, in this vein, makes me disagree for 2 reasons.

1) The ap227 lists AIM elements for piping_system, piping_system_line, and piping_system_line_segment as various subtypes of product_definition. These are:

a) piping_system
b) plant_line_definition
c) plant_line_segment_definition.

So, whether it is valid or not, the standard does define these as product_definition subtypes.

2) A plant line segment/line can be thought of as a physical product-it is the aggregate of its constituent plant_items, just as is the plant. You can go out into the plant and see line numbers on things.

However, you could also be right, by saying that these are abstractions. But so is a plant_item, which is an abstraction of a piece of material, etc.

Anyhow, thanks for the response. Hope that there are more.

Comment **Date:** 16-Dec-97 **Commentor:** Bill Burkett **Action:**

A piping_system_line and piping_system_line_segment are not product_definitions, because they are neither physical products nor are they products that are produced for sale. A piping_system_line is an abstract representation of the piping functionality of a plant and is, in that sense, part of the definition of a plant. Think of the piping_sytem_line as corresponding to the lines on a P&ID - they are not a product. They're not even pipes!

There is a relationship between a piping_system and piping_system_line. It is the assignement of the piping_system_component to a piping_system_line_segment.

This abstraction business can be confusing. I'm curious as to whether there are other interpretations among the members of this exploder.

Comment **Date:** 17-Dec-97 **Commentor:** Bill Burkett **Action:**

A few more thoughts:

>1) The ap227 lists AIM elements for piping_system, piping_system_line, and piping_system_line_segment as various subtypes of product_definition.
>These are:
>
> a) piping_system
> b) plant_line_definition
> c) plant_line_segment_definition.
>
>So, whether it is valid or not, the standard does define these as product_definition subtypes.

Evidentally I must have misinterpreted your original comment. You are right - these are subtypes of product defiinition.

But now I don't understand what your point is.

>2) A plant line segment/line can be thought of as a physical product-it is the aggregate of its constituent plant_items, just as is the plant. You can go out into the plant and see line numbers on things.

No, plant line and line segments *cannot* be thought of as physical product according to the AP 227 ARM. These terms are defined as functional objects - not physical objects. The plane line that you are referring to is the *realization* of the abstract plant line in terms of the components of a physical piping system.

>However, you could also be right, by saying that these are abstractions. But so is a plant_item, which is an abstraction of a piece of material, etc.

For good-or-bad, functional and physical plant items are treated different in AP 227 than functional and physical plant systems. Functional and physical plant items are differentiated with a functional/physical subtype/category - it's the same structure with two roles. Functional and physical plant systems are two completely different structures in the ARM. The reason for this - which you are free to accept or reject - is that the functional plant system structure was made overt in the model to more clearly show the relationship and role of a P&ID within AP 227. In addition, it provided a clearly identifiable subset of the ARM that directly overlapped with AP 221, so in this sense it fosters "harmonization" between the two APs. Since the functional system configuration was of primary importance in AP 221, and plant item of much lesser importance, separating out the functional plant system made sense, whereas separating out the functional plant item did not. (Incidentally, the sitation in AP 227 was just the opposite of AP 221: the physical plant item was of primary importance to the AP, whereas the functional system was of less importance (not unimportant, mind you - just less important).

Comment **Date:** 17-Dec-97 **Commentor:** Jay Roberts **Action:**

Within the context of plant_spatial_configuration, there doesn't seem to be a global parent for functional objects. However, these (and other) product_definitions are tagged as functional objects by product_definition::frame_of_reference (product_definition_context), where product_definition_context::name = 'functional occurrence' or 'functional definition'. This still leaves me with the question of whether system, line, segment are occurrences or definitions (I tend towards occurrences). Also, whether these product_definitions should have their own unique product entity or should they (via product_definition_formation) point to the plant product entity.

The NIST example is inconsistent on this-the various plant_system product_definitions (electrical_system, et al) have independent product entities, while the line and segment product_definitions (plant_line_definition, plant_line_segment_definition) provide definition to the plant product entity.

In lieu of a more global solution, my feeling would be to have all these product_definitions possess their own independent product definition. This would keep them from being bound to a particular product_definition_formation for the plant product entity, which could conceivable cause some problems with plant revisions. The mapping tables could easily be altered to reflect this and the necessity for the appropriate frame_of_reference value (they are currently silent on this issue).

I'm interested in hearing other comments on this.

Comment **Date:** 17-Dec-97 **Commentor:** Bob Fisher **Action:**

The inconsistencies pointed out do not surprise me: I suspect they arise from the fact that the functional/physical split which is a fundamental in AP221 was not a part of the original AP227 design, but retrofitted to try to attain AP221 compatibility.

Bill's interpretation is the correct approach (from where I stand and if AP221 compatibility is ever to be achieved). It is consistent with my view of the world and (I think) the EPISTLE view generally. If the existing AIM classifies these as product_definition, it appears that this was either overlooked in the retrofitting or is a symptom of something lacking in the integrated resources. Is there a global parent for functional objects, as opposed to products?

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 93 **Alt. Numbers:** IF-58 **Issue Date:** 18-Feb-98
Issue Title: Geometry Issues
Issue Owner: Nikolay Shulga **Country:**
Doc number SC4 N580 **Clause(s):**
Issue status: open **Classification**
Source: Implementer's Forum **Subject:**
Description:

Geometry issues can be divided into three categories:

1) CSG primitives that are vital to process plant industry. These are fully defined CSG primitives; they are not included in Part 42 now "because they can easily be constructed using a boolean". Our problem is that while constructing one pipe primitive out of a boolean is a minor nuisance, doing 2e5 of them is a major problem. This is a point that has to be put very strongly to Part 42 team until they go white and shake uncontrollably.

2) 'partially defined' shape representation, defined by a sequence of cross-sections. Most important in AP

227 context is square-to-round; also think of HVAC in general, or electrical cable boxes. Cross-sections are guaranteed to come in right; the rest is the responsibility of the importing CAD system. This should eventually become an AIC or something like that; we can do it within AP 227.. I think. An alternative - square-to-round primitive - won't pass Part 42 mavens.

3) Catalogue-based shapes: no explicit shape, the importing CAD system reconstructs it at its own risk. For now has to be handled with implementors' agreements; eventually to be resolved within PLIB

Unresolved - should be new entities in Part 42/2

HEMISPHERE
TRIMMED_SPHERE
?.?.? Truncated Sphere

A truncated_sphere is a CSG primitive. It may be envisioned as a boolean difference of a sphere and a half_space_solid (6.4.19) whose base_surface is a plane perpendicular to the Z axis passing through the point at Z axis at height disctance from the sphere position.location.

Express specification:

```
ENTITY truncated_sphere SUBTYPE OF sphere
height: length_measure;
WR1: (-radius < height < radius)
END_ENTITY
```

Attribute definitions:

height: the cutting plane is perpendicular to Z axis and passes through the point (0;0;height)

?.?.? tube

A tube is a CSG element which may be envisioned as a boolean difference of two cylinders of different radii, equal height and the same position parameter.

Express specification:

```
ENTITY tube SUBTYPE OF (geometric_representation_item)
position : axis2_placement_3d;
height: positive_length_measure;
external_radius: positive_length_measure;
internal_radius: positive_length_measure;
WR1: external_radius > internal_radius;
END_ENTITY
```

Attribute definitions:

position: The location and orientation of the placement axis for the primitive.
height: the size of the primitive in the Z direction
external_radius: The radius of the external cylindrical surface
internal_radius: The radius of the internal cylindrical surface

Proposal **Date:** 18-Feb-98 **Proposer:** Nikolay Shulga

What follows are my proposals on resolving item 1) from the issue description. Some of the proposals are based on Part42/2. NB: This edition has a new entity, faceted_primitive, which is essentially a back door into CSG. It is simply a list of points. An AP can add WRs to make pretty much anything out of it. I suggested using it for eccentric_pyramid.

- Comment on 7.4.17 - faceted_primitive

It needs a local coordinate system (obvious numerical stability considerations; using mapped_item instead is cumbersome).

Proposed:
Add new attribute:
position : axis2_placement3d;

- Present in Part 42/2 ISO/CD 10303-42:1997(E) ISO TC 184/SC4/WG12 N145

- TRIMMED_CONE
- ECCENTRIC_CONE
- ECCENTRIC_CYLINDER
Use Part 42/2 7.4.11

- CIRCULAR_ELLIPSOID
Use Part42/2 ellipsoid 7.4.20

- ECCENTRIC_PYRAMID
Use Part42/2 faceted_primitive 7.4.17, first 4 points are the base, second 4 points are the top.

- PYRAMID
- currently boolean_result. Use Part42/2 rectangular_pyramid 7.4.16

- REDUCING_TORUS
Use Part 42/2 reducing_torus 7.4.13 .

- TRIMMED_BLOCK
- currently boolean_result. Use Part42/2 right_angular_wedge 7.4.15

- TRIMMED_TORUS
- use Part 42/2 reducing_torus 7.4.13 with end_radius = start_radius.

Item 2) from issue description may be solved by using a generic shape_rep with three components:

```
entity list_of_cross_sections subtype of(rep_item)
cross_sections: LIST OF curve
WR: every curve in the list is closed
end entity;

curve; - extrusion path

axis2_placement3d - LCS
```

Item 3) from issue description will have to be resolved by catalogue-based people; I would (unhappily) support Panos' proposal for a set of implementors agreements, as I too, don't see a better way for now. I'd look to Panos for providing at least the first draft.

Comment **Date:** 18-Feb-98 **Commentor:** Mitch Gilbert **Action:**
Response to item 1 of issue description:

OK, but the process plant vendors need to do that. I think I have worked out a way around that through some subtyping and constraining in the AP 227 long form. I mentioned this breifly at the last implementor's forum, but we
could discuss it again at the next meeting.

Response to item 2 of issue description:

Again, what you suggest is quite doable, and we can use the same entity construction paradigm as I mentioned above (without details, I know) to do the square to round.

Response to Nilolay's item 2 proposed solution:

We can do it, but it won't look exactly like this. I'll go over the entity construction at the NIST meeting if you would like.

Response to Nikolay's item 3 proposed solution:

Can we discuss the detailed requirements for catalogue-based geometry? I am not sure that I understand those requirements, and I would like to understand them so I could possibly come up with a suggestion that would help. I know this will be a major DIS ballot issue, so we will wind up discussing it anyway.

Comment **Date:** 18-Feb-98 **Commentor:** Nikolay Shul **Action:**
Hi Jean-Christophe,

You are right. Thanks for catching this. I'd prefer using axis1_placement, something like this: (if we derive from the sphere, then we have to use centre attribute + direction, I'd rather use axis1_placement to be consistent)

ENTITY truncated_sphere SUBTYPE OF geometric_representation_item
position:axis1_placement
height: length_measure;
radius:positive_length_measure;
WR1: (-radius < height < radius)
END_ENTITY

Comment **Date:** 18-Feb-98 **Commentor:** Jean-Christop **Action:**
In the description of most geometrical primitives, we can see the location AND the orientation. To do that, these entities use axis1_placement or axis2_placement_3d in their definition. cartesian_point is only used when the entity has no "preferred" direction (as sphere). In your proposal for hemisphere, you should also define an orientation
using axis1_placement. If by inheritance, you already get a point, perhaps you only need to add a direction?

Comment **Date:** 19-Feb-98 **Commentor:** Bob Fisher **Action:**
I am not sure about the definition of the "tube". I would prefer a definition where the internal radius was specified as a wall thickness and was OPTIONAL (to be defaulted by the receiving system if not supplied). Absence of the wall thickness means that only the outer surface is being passed and no information about the wall thickness is being passed. This is perfectly valid and is the "normal" case. Better this than forcing a sending system to invent an artificial internal radius, which might be close enough to the outer radius to create interpretation problems in the receiving system.

Comment **Date:** 24-Feb-98 **Commentor:** Bob Fisher **Action:**
>From Nik's proposal:
---------quote---------
Also, boolean may be generalized to elliptical pipes (are there such things?).
---------unquote--------

Eliptical duct is used in HVAC. It is not uncommon, and is used where greater capacity than round is required in the same ceiling space (depth). If the second radius can be optional (if absent, circular is assumed) a generic eliptical tube seems like a good idea. If the second radius must be explicitly stated each time, it is probably better to have separate primitives, as the round is a 99% case.

----------quote----------
>Is cylinder solid or hollow - CSG primitives are solid. To transmit hollow shape, need to use either a boolean or a surface_rep.
----------unquote----------

Using a boolean creates the same problem as raised with the internal radius of the pipe. The exporting system is going to create a boolean shape to remove material which will create a "minimum-thickness" surface and system accuracy differences may create problems. It is more practical to let the receiving system create the boolean (if it needs to), but if this is not to be the approach, it is probably better to use surface rep for hollow cylinders and csg for solid (or where it doesn't matter). However, there is still a problem with this in that once imported, there is no way to know that the csg solids are "in reality" hollow (or at least, not defined to be solid).

I have to repeat a point I have made before, which is that it was a fundamental mistake in part 42 not to separate the so-called "csg" geometry definitions from their potential use to create a solid, or one of various combinations of surfaces. Hindsight is 20/20, but even at this point, I see no reason to leave out useful information (this csg cylinder or partial torus geometry represents a hollow uncapped surface) when it is simply additive and does not affect the existing data structure for the primitive. I can also see that the ability to process this additional information might well be the basis of a conformance class. Individual APs might be able to store this qualifier also, but that would be logically the wrong place: the geometry engines should deal with it generically, eventually.

It needs to be accepted by the mechanical people that part 42 (Geometry definition) is not intended to be used just for defining shapes to manufacture to (is it??) but also for passing "envelopes" which have no implied concept of either solidity or hollowness, and we want to have the convenience and compactness of "csg-type" definitions to do this.
This is the real issue here. Until it is addressed, this issue will not go away.

This problem has a common element with the new cross-section transition element proposed, in that they both represent real-world requirements to pass partially defined shapes, although the actual geometry defined is complete and explicit. In the first instance, we are not attempting to define the surface connecting the cross sections and in the second, we are not attempting to say anything about what is or is not inside the envelope (hollow/solid). The problem is that with current csg conventions, it is ASSUMED to be solid and we have no way to say otherwise. Geometry and its use need to be separated (in the new architecture!)

A suggestion (having realised the nature of this assumption) is to have an optional (for forward compatibility) parameter for all csg primitives to identify their intended interpretation. Absence implies interpretation as a solid. If present,
S = solid, C = completely enclosed (capped) surface, A,B etc. = for surfaces which partially enclose the volume and whose meaning will depend on the particular primitive. Use of

letters avoids confusion with an optional dimensional parameter.

Comment **Date:** 24-Feb-98 **Commentor:** Nikolay Shul **Action:**
Final (hopefully) summary.

Part 42/2 primitives:
Apparently everybody is happy with these

Pipe primitive:
Proposal from Bob F. - use thickness instead of inner radius. To me one is no more artificial than other, I don't care either way. Comment from Hiroshi M. - pipe primitive is unnecessary, processing large numbers of these should not be a problem. Also, boolean may be generalized to elliptical pipes (are there such things?). I am not sure about this.. it's true the boolean can be kludged (er, special-cased) in most cases, so the penalty may be minimized. OTOH, I feel pipe shape is so common that it deserves to be a primitive.

Truncated sphere - I think the same comment from Hiroshi M., my comments apply as well.

Square-to-Round - use partially defined shape rep - a sequence of cross-sections. I think Mitch has some ideas on this.

Catalogue-based shapes - an implementor's agreement is the only practical approach, a draft from catalogue-based people would help.

Is cylinder solid or hollow - CSG primitives are solid. To transmit hollow shape, need to use either a boolean or a surface_rep.

I'd like to close the CSG part on 17th.. at least for the time being.

Comment **Date:** 25-Feb-98 **Commentor:** Mitch Gilbert **Action:**
At 08:55 AM 2/25/98, Nikolay Shulga wrote:
>Hiroshi et al,
>
>Here's the problem as I see it: if you want to use CSG operations, you have to make a CSG operand out of extruded_solid. The only way to do it is to make a representation out of it.
>

Is not extruded_solid a subtype of solid_model. If it is, then it is capable of being used as a boolean operand. The idea that I had for the CSG primitives that AP 227 cannot convince the Part 42 team to include consists of making a common subtype of solid_model and shape_representation for each of the primitives, constraining the contents of the items set of representation to hold the parameters that define the Plant specific primitives. We can discuss this design at the workshop next month.

>I think the real problem is that we are not really doing CSG - only unions. So our requirements to our CSG primitives are a lot more relaxed than Ray G.'s requirements to his CSG primitives. I don't think these can be reconciled - Part 42 CSG as it is now has certain requirements, our 'relaxed CSG' also have certain requirements, and these do not intersect well. I think a long term solution is to have a 'relaxed CSG' class.
>

Like I said, using my design above, concocting our own more loosely defined CSG primitives specifically for use in the plant design context is not out of the question.

Comment **Date:** 25-Feb-98 **Commentor:** Hiroshi Mura **Action:**
You've forgot to see totally that sweeping along a path creates a pipe more effectively than CSG, and the Part42/v2 is equipped with such an insturument, including the elliptical ones.

Comment **Date:** 26-Feb-98 **Commentor:** Hiroshi Mura **Action:**

My comment is interleaved with quotes in the following.

> Here's the problem as I see it: if you want to use CSG operations, you have to make a CSG operand out of extruded_solid. The only way to do it is to make a representation out of it. (by NS)

> Is not extruded_solid a subtype of solid_model. If it is, then it is capable of being used as a boolean operand. The idea that I had for the CSG primitives that AP 227 cannot convince the Part 42 team to include consists of making a common subtype of solid_model and shape_representation for each of the primitives, constraining the contents of the items set of representation to hold the parameters that define the Plant specific primitives. We can discuss this design at the workshop next month. (by MG)

I think what Mitch wrote has effectively answered the question of NS. But I would like to clarify:
In Part42, manifold_solid_brep, swept_face_solid, swept_area_solid are all sub-entities of solid_model, and a solid_model can be an operand of a Boolean operation. You can make a Boolean between a csg_primitive(it's
also a subentity of solid_model) and a solid_model.

Also, there's a half_space_solid to be used as an Boolean operand to cut 'csg_primitive's with a 'plane' for example to make a partial sphere or torus, etc.

More importantly, again and again as I said that you can make a tube including bent ones, with a sweeping. For this, you have revloved_area_soild, extruded_area_solid, revolved_face_solid, extruded_face_solid with Part42 at presnet. With Part42/v2 you have in addtion, plannar_swept_face _solid, and planar_swept_face_solid, both of
which can sweep a face WITH A HOLE along any planar path. You have a tube without recourse to Boolean operation. Probably there's more than just a tube you can make with them and I cannot see why AP227 team need a
limited use contruct like 'tube' at all?! You need also a hollow torus, hollow block, and first of all, hollow sphere to make an oil reservoir. These shape necessities are not solved by adding a limted use primitve to csg_select and adding only a single tube to the selection only comlicate the mapping.

I feel frankly that the present AP227 team has not exploited well the Part42 even as is, before any revisions or additions. You must not judge things by their names. For example, how it is easy to creat a faceted csg shape within Part42, though its name is 'faceted_brep' and it can be used as an operand of Boolean operation.(I myself once requested "faceted_csg" and withdrew) You can make hemisphere or "demi ellipsolid" easily with a combination of a relevant primitive and a half_space_solid in this case 'plane', and it is more generic than introducing "partial_primitive"
options and quite easy(if people understand the structure).

> I think the real problem is that we are not really doing CSG - only unions. So our requirements to our CSG primitives are a lot more relaxed than Ray G.'s requirements to his CSG primitives. I don't think these can be reconciled - Part 42 CSG as it is now has certain requirements, our 'relaxed CSG' also have certain requirements, and these do not intersect well. I think a long term solution is to have a 'relaxed CSG' class. (NS)

>>I think the real problem is that we are not really doing CSG - only unions. So our requirements to our CSG primitives are a lot more relaxed than Ray G.'s requirements to his CSG primitives. (MG)

I summerized this characteristic of process plant geometry treatment with a powerpoint presentation in Florence meeing in a Part42-processplant-builidng joint meeting, using the term, 'primtive-plileup' method. I hope you already have a copy.

I think acturally there are many solutions to the problem the AP227 metioned as "voids" in Part42 or in other IRs. If you regard the primitive pileup or 'loose csg' as a representation with unevaluated superpositions, then it is a
"representation" and you can place the shapes under 'representation' with a representation _context attached, or you can name the 'representaiton' as such.

It is impossilbe though, as far as present AP227 ARM-AIM mapping requires that one primitive lies under one representation, as I quote in the follwing the remark by J R:

(At 2:51 PM -0500 11/19/97, Jay Roberts wrote:
> According to the DIS version of AP227, multiple instances of shape_representation_element_usage may be associated with one shape_representation in the ARM.
>
> However, it is impossible to represent this situation in the AIM due to the nature of the AIM elements for the ARM entities.
>
> The AIM element for shape_representation is PROPERTY_DEFINITION_REPRESENTATION. The AIM element for shape_representation_element_usage is REPRESENTATION or one of it's various subtypes.
>
> The relationship between the two in the ARM is manifested in the AIM as:
>
> PROPERTY_DEFINITION_REPRESENTATION.DEFINITION = REPRESENTATION.
>
> Hence, a given instance of shape_representation in the ARM can only be related to a single instance of shape_representation_element_usage.

The Part41,42 and 43 combination DOES NOT require that a single csg primitive must exist under repesentaion. On the contrary, it encourage such a use; the idea behind this is to group together the component shapes of an object under one representaion in the field of 'items[1:?] of the entity. Only the ARM and mapping of the AP227 preclude the effective use of this. This is a vital error. A file created by ARM-AIM translation method is more than 100 times bigger
than direct proprietary CAD to AIM translation in pcals experiment.

I feel rather AP227's shape related ARM defintions require a lot of straightening

Resolution: **Res Date:** **Impl Resp:**

Issue Number: 94 **Alt. Numbers:** IF-59 **Issue Date:** 19-Feb-98
Issue Title: Geometry - Hollow Tubes
Issue Owner: Pano Polihroniadis **Country:**
Doc number SC4 N580 **Clause(s):**
Issue status: open **Classification**
Source: Implementer's Forum **Subject:**

Description:

I am not sure if the following issue has been addressed with AP227.

May be the following is more of a conformance class issue rather than we need to actually do something with AP227.

Although tubes, elbows etc are by nature hollow some CAD systems, including PDMS, hold these in the main as solid, ie closed cylinders and closed torii. PDMS is able to hold these as hollow but it has been accepted for simplicity reasons and fast drawing to hold these as solid. In fact, in PDMS, as most of our objects are drawn through Catalogues, the User, if he wishes to, can set different geometric representations for the same thing, therefore he could draw it either as hollow or solid.

When I transfer geometry from other systems, sometimes I have to make a decision if I can make an 'intelligent' guess, and substitute what I am given with what I need to store in PDMS. For example if I am getting a hollow cylinder when I know is a tube then I could make it a solid cylinder etc.

If this aspect of 'hollowness' has not been addressed yet, now is a good time as we are thinking about geometry again.

I have the following comments on the subject:

Schematic or Real Geometry

I believe, it is not very clear, if the geometry we are exchanging is a schematic representation of the object or the 'real' thing with all its accuracy. It all depends what this geometry is used for. For example, in one level the external envelop shape is necessary so that clashes with other parts can be looked at. On the other hand if one wants to find the weight of an object or one needs to set attributes to the internal surfaces to objects then one needs the 'real' geometry. How real is real? I certainly believe that we do not need full details as in the mechanical industry. On the other hand we just finding out that in PDMS we have to deal with welding attributes etc which implies that one has to poit to individual surfaces. This is a difficult subject. May be what we need to say something in AP227.

Instantiation of an object

If we were to accept that when we exchange geometry we may need to give more that one representation for the same object, for the reasons outlined above. In this case, it makes more sense to define the geometry of an object once and when we instanciate it we only need the transformation without the repitition of the geometry. I realise that this facility is already there but is a matter of Implementors agreement of what we are exchanging and I know sometimes we bend these rules!

Hollowness

In more simplistic terms may be we need to put an attribute in AP227 to indicate to the implementors that some geometry, like a tube and an elbow could be substituted by a solid by the receiving system. Are there any systems out there that can not do hollow tubes anyway?

and finally to the old chestnut!
Catalogues

At the end of the day we, in PDMS, and I believe the same applies to PDS and other systems, we have to substitute the geometry of the object we read from an AP227 step file to the one which is stored in our catalogues, therefore we may lose the geometry found in the step file and what this implies etc

Comment **Date:** 19-Feb-98 **Commentor:** Bob Fisher **Action:**

>I am not sure if the following issue has been addressed with AP227.
>
>May be the following is more of a conformance class issue rather than we need to actually do something with AP227.
>
>Although tubes, elbows etc are by nature hollow some CAD systems, including PDMS, hold these in the main as solid, ie closed cylinders and closed torii. PDMS is able to hold these as hollow but it has been accepted for simplicity reasons and fast drawing to hold these as solid. In fact, in PDMS, as most of our objects are drawn through Catalogues, the User, if he wishes to, can set different geometric representations for the same thing, therefore he could draw it either as hollow or solid.
>
>When I transfer geometry from other systems, sometimes I have to make a decision if I can make an 'intelligent' guess, and substitute what I am given with what I need to store in PDMS. For example if I am getting a hollow cylinder when I know is a tube then I could make it a solid cylinder etc.
>
>If this aspect of 'hollowness' has not been addressed yet, now is a good time as we are thinking about geometry again.
>
>
>I have the following comments on the subject:
>
>Schematic or Real Geometry
>---
>I believe, it is not very clear, if the geometry we are exchanging is a schematic representation of the object or the 'real' thing with all its accuracy. It all depends what this geometry is used for. For example, in one level the external envelop shape is necessary so that clashes with other parts can be looked at. On the other hand if one wants to find the weight of an object or one needs to set attributes to the internal surfaces to objects then one needs the 'real' geometry. How real is real? I certainly believe that we do not need full details as in the mechanical industry. On the other hand we just finding out that in PDMS we have to deal with welding attributes etc which implies that one has to poit to individual surfaces. This is a difficult subject. May be what we need to say something in AP227.

rjf>>> In the old ARM diagram i have, plant_item_shape can be multiply identified with one or more of three classifications:

envelope/outline/detail. The principle is OK. The number of options may be insufficient and/or too vaguely defined and I assume this is the discussion topic, unless something changed while I was out of the loop?

The intended usage of the geometry to be passed definitely needs to be part of the conformance class discussions.

I think the point about welding attributes needs more discussion as it is not at all obvious to me that this requires a pointer into the graphics.

On how real is real? There is a complete ISO New Work Item (Dimensional Inspection) dedicated to classifying and documenting the 7 or 8 degrees of accuracy they have identified in the measurement of physical dimensions and surface properties. I suspect our discussion actually >adds some levels to that at the macro end.<<<
>
>Instantiation of an object

>-------------------------
>If we were to accept that when we exchange geometry we may need to give more that one representation for the same object, for the reasons outlined above. In this case, it makes more sense to define the geometry of an object once and when we instanciate it we only need the transformation without the repitition of the geometry. I realise that this facility is already there but is a matter of Implementors agreement of what we are exchanging and I know sometimes we bend these rules!

rjf>>> I don't understand this point. If the geometry for multiple representations is identical, it can be pointed to twice. If it is not identical, it must be instanced separately, anyway. I don't know if this is related to the hollow/solid discussion, but I can't think of a situation where two similar geometries would be passed through, one hollow and one solid. Or maybe by different representation, you meant csg and b-rep? I assumed different representation meant "detailed envelope" and "interference envelope".<<<

>
>Hollowness
>----------
>In more simplistic terms may be we need to put an attribute in AP227 to indicate to the implementors that some geometry, like a tube and an elbow could be substituted by a solid by the receiving system. Are there any systems out there that can not do hollow tubes anyway?

rjf>>> I fully agree a hollow/solid attribute is necessary. I have made the same point myself in previous email. Logically, there are three options: hollow open ended, hollow capped and solid.<<<

>
>and finally to the old chestnut!
>Catalogues
>----------
>At the end of the day we, in PDMS, and I believe the same applies to PDS and other systems, we have to substitute the geometry of the object we read from an AP227 step file to the one which is stored in our catalogues, therefore we may lose the geometry found in the step file and what this implies etc.

rjf>>> "Catalog" items can mean
1) things stored in a catalog in PDS or PDMS or .....
2) things which are highly standardised with a generally available and agreed definition.

These definitions are a long way from synonymous. (2) is a subset of (1) and I would strongly suggest we use it and restrict the scope of this discussion to standard piping components and similar. The first point of business is to agree the scope.

There are three possible modes for transfer of "catalog items" a) transfer a short code which we agree represents component xyz b) transfer a group of properties which are sufficient to identify a matching component in the catalog of the destination system c) transfer full details (including geometry) and instantiate a catalog entry on-the-fly on receipt of the data.

(c) is problematic in a lot of systems, but restriction of the scope above to (2) should make it not necessary, as the necessary catalog items can be put in place in advance. Extent of items in the catalog of the destination system may need to be part of a conformance class definition.<<<

Comment **Date:** 20-Feb-98 **Commentor:** Hiroshi Mura **Action:**
Following is my comment.

>>Instantiation of an object
>>-------------------------
>>If we were to accept that when we exchange geometry we may need to give more that one representation for the same object, for the reasons outlined above. In this case, it makes more sense to define the geometry of an object once and when we instanciate it we only need the transformation without the repitition of the geometry. I realise that this facility is already there but is a matter of Implementors agreement of what we are exchanging and I know sometimes we bend these rules!

> rjf>>> I don't understand this point. If the geometry for multiple representations is identical, it can be pointed to twice. If it is not identical, it must be instanced separately, anyway. I don't know if this is related to the hollow/solid discussion, but I can't think of a situation where two similar geometries would be passed through, one hollow and one solid. Or maybe by different representation, you meant csg and b-rep? I assumed different representation meant "detailed envelope" and "interference envelope".<<<

It is not necessary even under the status quo of the IRs, the geometries to be instanciated are the same: Thing is that there are built-in transformation mechanism defined in Part43 that can modify the 'scaling' of each object, in which the sense of scaling is a little bigger than just a multiplication or a rotation/translation of coordinate values. You can apply a full transformation matrix to change shape before the base shape is actually instantiated. In this sense, what Panos says is correct. Topology is the same, but geometries are not the same between 2 instantiations.

A bigger problem is that at present, because of a crucial loophole of ARM to AIM mapping, in practice, as once Jay ROBERTS wrote, each shape must consist of one geometric element, say for example, you cannot define a pump
as a combination of 4 cylinders, 1 cone, 2 blocks, etc., ... Instead, each cylinder, each cone, each block must have a definitional coordinate system typcially reprsented by an axis2_placment3D, with which they make a set of representation_item within a 'representation'. Conversely each block, each cylinder, each cone must have an entity 'representation'. It is a fatal loss of efficiency. Moreover, in essence, it does not define the GEOMETRY for the pump. It is just a loose collection of geometry pieces, sematincally grouped together for a 'product'. So we cannot apply a
tranformation matrix to this collection as a unit: The transformation can be applied defining a target and a source representation.

This problem is not at the scale of 2 or 3 lines difference in favour of genericity to define a tube with 2 cylinders: If we follow the recommendation suggested by Jay for a quick fix, a whole bunch of representation, property_definition_reprentation, product_definition, etc, and more of a long sequence, you need define for each block, each cylinder..... Moreover, it cannot receive a correct geometries defined in other APs: It lacks the interoperability with other APs in the core of geometry definition per se. Namely, any number of pieces of CSG or other
geometry elements under one representation to form THE SHAPE of the object. in wich case you can still apply a transformation once to create an transformated instance. (This comment is not very much elaborated in my
DIS commets, for a short of time.)

In reference to Catalog, if you define the geometry of a catalog as such a combination of geometry under one representation, you can still change the length of a pipe, for example.

In conclusion, we need to restructure the ARM (and its mapping to AIM)of the AP.

Resolution: **Res Date:** **Impl Resp:**